Analytic and numerical aspects of isospectral flows

Amandeep Kaur

Wolfson College

DAMTP, Center for Mathematical Sciences
University of Cambridge

April 2016

Abstract

In this thesis we address the analytic and numerical aspects of isospectral flows. Such flows occur in mathematical physics and numerical linear algebra. Their main structural feature is to retain the eigenvalues in the solution space. We explore the solution of Isospectral flows and their stochastic counterpart using explicit generalisation of Magnus expansion.

In the first part of the thesis we expand the solution of Bloch–Iserles equations, the matrix ordinary differential system of the form $X' = [N, X^2], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n), \; N \in \mathfrak{so}(n)$, where $\mathrm{Sym}(n)$ denotes the space of real $n \times n$ symmetric matrices and $\mathfrak{so}(n)$ denotes the Lie algebra of real $n \times n$ skew-symmetric matrices. This system is endowed with Poisson structure and is integrable. Various important properties of the flow are discussed. The flow is solved using explicit Magnus expansion and the terms of expansion are represented as binary rooted trees deducing an explicit formalism to construct the trees recursively. Unlike classical numerical methods, e.g. Runge–Kutta and multistep methods, Magnus expansion respects the isospectrality of the system, and the shorthand of binary rooted trees reduces the computational cost of the exponentially growing terms. The desired structure of the solution (also with large time steps) has been displayed.

Having seen the promising results in the first part of the thesis, the technique has been extended to the generalised double bracket flow $X' = [[N, X] + M, X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n)$, where $N \in \mathrm{diag}(n)$ and $M \in \mathfrak{so}(n)$, which is also a form of an Isospectral flow. In the second part of the thesis we define the generalised double bracket flow and discuss its dynamics. It is noted that $N = 0$ reduces it to an integrable flow, while for $M = 0$ it results in a gradient flow. We analyse the flow for various non-zero values of N and M by assigning different weights and observe Hopf bifurcation in the system. The discretisation is done using Magnus series and the expansion terms have been portrayed using binary rooted trees. Although this matrix system appears more complex and leads to the tri-colour leaves; it has been possible to formulate the explicit recursive rule. The desired structure of the solution is obtained that leaves the eigenvalues invariant in the solution space.

Keywords: Isospectral flow, ordinary differential equations, eigenvalues, Magnus expansion, Lie group, Lie algebra, binary trees, Cayley transform, double bracket flow, Toda lattice, QR algorithm, Runge–Kutta.

Dedicated to my late grandmother, Veeran Wali.

Acknowledgements

I never thought that writing the acknowledgements section in my thesis would be more difficult than any other section. I express the deepest appreciation to each and everyone who helped me in these four years at Cambridge.

First of all, I would like to express my gratitude to my supervisor Arieh Iserles for giving me the opportunity to work in DAMTP and to pursue research in Mathematics in such a beautiful University. He truly is a great mathematician and his suggestions have always been valuable to me. Also, my adviser Carola-Bibiane Schönlieb has always been a great support. I thank her for motivating me from time to time.

My officemates and friends Evangelos Papoutsellis and Pranav Singh created a beautiful working environment. I have learnt a lot having both of them as my friends. Evangelos and I had an exchange program for teaching me Greek in exchange of every Hindi(or Punjabi) word he learnt!

I thank all the academic and supporting staff at the department for being helpful in various ways. I acknowledge all the visitors during my stay at Cambridge; professors, postdocs, and students, the discussions with them were highly knowledgeable and fruitful.

I sincerely thank Wolfson college for accepting me as a graduate student and for providing me with accommodation for four years. The porters and all other staff have been very helpful to make this a home away from home. I appreciate my friends Zab, Varun, Rav, Angad, Deepyanti, Liza, Adarsh for all the fun and support, and the residents of Toda house for being so understanding.

A special word of gratitude is due to Nuri Purswani for all her support. She has been a great friend and stood with me in all the tragic moments I had to go through during my time at Cambridge. I also thank my friends Parminder Kaur and Ankur Dutt for their support, specially in the writing period. I am indebted to Mugdha who allowed me to stay at her place.

I acknowledge Shiromani Gurdwara Prabandhak Committee–Cambridge Commonwealth Trust (SGPC–CCT) for the financial support. Also, I am grateful to my friend Nuri my brother Davinder who contributed to the funding in the final year of my PhD.

I dedicate this thesis to my beloved grandmother who died just before I started my journey to Cambridge.

I am indebted to my cute nephews Jaiveen, Ganeev and Fateh for making me smile all the time. I thank Shavinder and Nishu (my sisters-in-law) for their love and support.

This thesis would have been impossible without the support of my loving parents Jaspal Singh and Kulwant Kaur and my brothers Davinder and Kuljinder. Their unconditional love, patience and support in every aspect help me to move forward. I thank my brothers for being there at every single step helping me cross all the hurdles. A special thanks to my parents for nurturing me. They have always been a source of encouragement and inspiration.

And I am greatful to someone special, my friend, guide and protector, for always being with me... *So Satgur pyara mere naal hai*!!!

Statement of Originality

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text

Contents

Chapter 1

Introduction

Isospectral flows are matrix systems of ordinary differential equations of the form

$$X^{'} = [B(X), X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n), \tag{1.1}$$

where $B(X) : \mathrm{Sym}(n) \to \mathfrak{so}(n)$. Their main structural feature is that they preserve the eigenvalues of the solution matrix. Isospectral flows occur in many important applications. First and the best known example is the Toda lattice, a one-dimensional lattice of particles whose motion is described by a nearest-neighbour interaction of an exponential type. It can be used to model a wide range of particle systems, ranging from the hard-sphere limit to the atomic case [Tod81, Tod89]. Another important example is the QR flow. The QR method for finding the eigenvalues of a matrix can be executed as an isospectral flow at unit intervals. QR flow is the generalization of non-periodic Toda flow. Such flows were first investigated by Symes [Sym82] and subsequently in [Nan82, DNT83, DRTW91, Chu84, CD89, Lag91] and elsewhere.

Other well known examples include *eigenvalue problems* and *inverse eigenvalue problems for symmetric Toeplitz matrices* [FNO87].

Note that if we let $B(X) = [N, X]$, in (1.1) where $N \in \mathrm{Sym}(n)$ then it leads to the *double-bracket flows*. Double bracket flows are isospectral flows given by the equations

$$X^{'} = [[N, X], X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n) \tag{1.2}$$

where $N \in \mathrm{Sym}(n)$. They were introduced by Brockett [Bro91] and Chu and Driessel [CD90]. Double bracket flows were discretised and then solved by Iserles by the method of Magnus series [Ise02] and were generalized for more parameters [BI05]. Also, methods based on Magnus expansion are proposed for the numerical integration of the double-bracket flow and a bound on the convergence domain is provided by Casas [Cas04].

It is obvious that we can discretise isospectral flows by traditional numerical methods (e.g. Runge–Kutta and multistep), but, once $n \geq 3$, these methods cannot respect the isospectrality of the system, i.e. the numerical solution changes the eigenvalues [CIZ99]. Isospectrality is essential for applications ranging from classical mechanics, like Toda flows and N-body systems, to linear algebra, like QR flows and inverse eigenvalue problems, so we need to solve systems of the form (1.1) by a method that respects it [CIZ97, CIZ99, Zan98].

In this thesis we solve the given isospectral flows (namely, Bloch–Iserles equations (3.1) and generalised double bracket flow (4.1)) using the method of Magnus series. For the solution, first we convert the isospectral flow to a Lie-group flow and then translate it into a Lie-algebraic equation. This method preserves the isospectrality and gives the desired structure of the solution with large time steps. We show that the solution of an isospectral flow can be represented in the form $X(t) = e^{\Omega(t)}X_0e^{-\Omega(t)}$, where instead of computing X at the first place, we obtain the Taylor expansion of Ω. Note that this ensures automatically that the numerical solution, being similar to X_0, is isospectral. We will see that the Taylor expansion of Ω can be formed algorithmically from the elements of the flow and linear combinations of their commutators and(or) anti-commutators. Our goal is to determine the rules for finding the terms of Ω to an arbitrary accuracy. The terms are represented by binary rooted trees and an algorithm is formed to construct the next tree by recursion and to calculate the coefficient of each tree. This lays the foundations to a more general setting, namely the explicit representation of the solution when $B(X)$ can be represented in a finite "alphabet". The representation as binary trees is very important because otherwise, as the number of terms in each iteration grows exponentially, the complexity of manual computation becomes prohibitive. By indexing the terms in the expansion with a subset of binary trees, it is convenient to derive explicit recurrence relations. Also, it is remarkable that the skew-symmetry and Jacobi identity obeyed by the commutator help us to reduce the number of the terms by cancelling or writing certain terms as the linear combination of other terms.

We organise the thesis as follows. In chapter 2, we discuss the theory of isospectral flows and their applications ranging from classical mechanics to linear algebra. We also discuss the classical numerical methods, isospectral methods, and give introduction to Lie group and Lie algebra along with illustrating some examples. Towards the end of the chapter, we explain Magnus expansion which is used in the later chapters for discretisation, and define some basic concepts of graph theory for constructing the binary rooted trees for the expansion terms.

Chapter 3 consists of expansion of the solution of Bloch–Iserles equations, a matrix ordinary differential system of the form $X' = [N, X^2], \ t \geq 0, \ X(0) = X_0 \in \mathrm{Sym}(n), \ N \in \mathfrak{so}(n)$, where $\mathrm{Sym}(n)$ denotes the space of real $n \times n$ symmetric matrices and $\mathfrak{so}(n)$ denotes the Lie algebra of real $n \times n$ skew-symmetric matrices. We discuss the motivational properties of the system. This system is an isospectral flow, and it has been seen that the system is endowed with Poisson structure. We represent the solution of isospectral flow in the form $X(t) = e^{\Omega(t)} X_0 e^{-\Omega(t)}$, where instead of computing X at the first place, we obtain the Taylor expansion of Ω. The numerical solution being similar to X_0, ensures the isospectrality. The solution is computed using Magnus series and the Taylor expansion is formed from the elements X_0 and N, and the linear combination of their commutators and anti-commutators. We also introduce curly bracket, $\{X_0\} = NX_0 + X_0N$, that means the matrix N features implicitly in the bracket. Thus, each term of the expansion is made of commutators and curly brackets with some expression X_0. This representation further contributes to the shorthand of binary rooted trees by preventing bicolour leaves. It has been shown graphically that the solution with the Magnus series preserves the spectrum of the solution space. The terms of the expansion are represented using binary rooted trees that helps to reduce the manual computation which has been growing exponentially with each iteration. Moreover, we deduce an explicit formalism to construct the trees and their coefficients recursively. Results are analysed by plotting the error graphs of the solution and the graphs for error in eigenvalues, these plots are generated in MATLAB.

In chapter 4, we introduce the generalised double bracket flows, the matrix system of ordinary differential equations of the form $X' = [[N, X] + M, X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n)$ where $N \in \mathrm{diag}(n)$ and $M \in \mathfrak{so}(n)$. Analysis is done for 2×2 and 3×3 matrices and the dynamics are represented as phase portraits. We see that the phase plots, computed with MATLAB result in spirals. It is observed that the system undergoes a Hopf bifurcation that gives birth to the limit cycles. We further discuss the special cases where we see that the different weightage of N and M influence the result. We also discuss the case with identical diagonal elements of the matrix N. We perform these experiments taking 3×3 matrices.

In chapter 5, we extend the method of Magnus series, to generalised double bracket flow. This system seems to be more challenging to be discretised and for the solution to be deduced as an explicit representation. Due to the fact that it is going to have three matrices appearing implicitly in the Taylor expansion together with commutators. We obtain the Taylor expansion and represent the terms as binary rooted trees. Although this matrix system appears more complex and leads to the tri-colour leaves; it has been possible to formulate the explicit recursive rule. In the case of this matrix system, we have two types of trees and an interplay between them, which makes it more complicated. A step by step algorithm is developed and recurrence formula is defined that enables us to compute the trees and coefficients explicitly. We compare the solution and error in eigenvalues against ode45 method. In the concluding chapter 6, we briefly summarise the work presented in this dissertation.

Chapter 2

Mathematical Preliminaries

In this chapter we assemble and briefly explain, the mathematical objects and the keywords that form a basis for the work presented in this dissertation. We start with explaining the mathematical term, isospectral flow in section 2.1 along with its applications and discuss about the classical methods and isospectral methods. This is followed by the introduction of Lie groups, Lie algebras, and their examples in section 2.2 and section 2.3 respectively. Later in section 2.4 we explain the Magnus expansion and the basic definitions of graph theory for the formation of binary rooted trees. Main references to this chapter are [IMKNZ99] [Ise99] [Zan98].

2.1 Isospectral Flows and their applications

Isospectral flows are the matrix system of ordinary differential equations of the form

$$X' = [B(t, X), X], \quad t \geq 0 \quad X(0) = X_0, \tag{2.1}$$

where $X, B(t, X) \in \mathbb{V}^{n\times n}$, set of $n \times n$ matrices with the entries in $\mathbb{V} = \mathbb{R}$ or $\mathbb{C}$. The functions B and X which satisfy the matrix differential equation (2.1) are called a *Lax pair* [Lax68] and $[A, B] = AB - BA$ is the commutator or *Lie bracket*. It is easy to verify that

$$X(t) = Q(t)X_0Q(t)^{-1}, \; t \geq 0, \tag{2.2}$$

is the solution of (2.1), where $Q(t)$ is the solution of

$$Q' = B(t, QX_0Q^{-1})Q, \quad t \geq 0, \quad Q(0) = I. \tag{2.3}$$

To verify this isospectral deformation: We begin with associating the differential equation

$$Q' = B(t, X)Q, \quad t \geq 0, \quad Q(0) = I, \tag{2.4}$$

where I is the identity matrix, to (2.1). Let us construct the matrix function $Q(t)^{-1}X(t)Q(t)$, we observe that, from (2.1) and (2.4), together with

$$\frac{d}{dt}Q^{-1} = -Q^{-1}Q^{'}Q^{-1}, \tag{2.5}$$

we get

$$\frac{d}{dt}Q(t)^{-1}X(t)Q(t) = O \tag{2.6}$$

where O is the zero matrix. Thus, we can say that $Q(t)^{-1}X(t)Q(t)$ is time independent. Therefore, it must equal its initial condition substituting $t = 0$. Hence,

$$X(t) = Q(t)X_0Q(t)^{-1} \tag{2.7}$$

is a similarity transformation that leaves the eigenvalues same. In the following chapters, while discussing about Bloch–Iserles equations and double bracket flows, we shall see that there are many important examples where B is skew-symmetric. Then the equation (2.4) becomes an *orthogonal flow*. In the case of *orthogonal flow*, $Q^{-1} = Q^T$ and if X_0 is symmetric, $X(t)$ also remains symmetric.

Isospectral flows occur in various applications. First and a well known example is the Toda lattice.

Toda lattices, introduced by Morikazu Toda (1967) are one-dimensional lattices of particles whose motion is described by a nearest-neighbour interaction of an exponential type and can be used to model a continuum of flows, ranging from the hard-sphere limit to the atomic case [Tod81, Tod89].

Consider the equations of motion

$$m\frac{d^2x_i}{dt^2} = \frac{d}{dr}\psi(x_{i+1} - x_i) - \frac{d}{dr}\psi(x_i - x_{i-1}), \;\; i = 1, 2, ..., \tag{2.8}$$

where m is the mass of all the particles, x_i is the displacement of the ith particle and $\psi(r)$ is the interaction potential. Introduce the momenta $p_i = mx_i^{'}$ and generalised coordinates $q_i = x_i, i = 1, 2, ...,$ and let ψ be the exponential potential

$$\psi(r) = e^{-r} + r, \tag{2.9}$$

such that $\psi^{'}(r) = -e^{-r} + 1$, the equations of motion become the Hamiltonian system

$$q_i^{'} = \frac{1}{m}p_i,$$

$$p_i^{'} = e^{-(q_i - q_{i-1})} - e^{-(q_{i+1} - q_i)} \tag{2.10}$$

assume $m = 1$ the corresponding Hamiltonian function is

$$H(\mathbf{p}, \mathbf{q}) = \frac{1}{2}\sum_i p_i^2 + \sum(e^{-(q_{i+1}-q_i)} - 1), \tag{2.11}$$

where $\mathbf{p} = \{p_i\}$ and $\mathbf{q} = \{q_i\}$.

To change this Hamiltonian system into isospectral form we introduce the change of variables as done by [Tod81] [Fla74]

$$\alpha_i = \frac{1}{2}e^{-(q_{i+1}-q_i)/2}, \tag{2.12}$$

$$\beta_i = p_i, \tag{2.13}$$

so that

$$\alpha_i^{'} = \alpha_i(\beta_i - \beta_{i+1}), \tag{2.14}$$

$$\beta_i^{'} = 2(\alpha_{i-1}^2 - \alpha_i^2), \tag{2.15}$$

we get the Lax form, $X^{'} = [B, X]$, where

$$X = \begin{bmatrix} \beta_1 & \alpha_1 & 0 & \dots & 0 \\ \alpha_1 & \beta_1 & \alpha_2 & \ddots & \vdots \\ 0 & \alpha_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \alpha_{d-1} \\ 0 & \dots & 0 & \alpha_{d-1} & \beta_d \end{bmatrix},$$

and

$$B = \begin{bmatrix} 0 & -\alpha_1 & 0 & \dots & 0 \\ \alpha_1 & 0 & -\alpha_2 & \ddots & \vdots \\ 0 & \alpha_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha_{d-1} \\ 0 & \dots & 0 & \alpha_{d-1} & 0 \end{bmatrix}. \tag{2.16}$$

Another important example is the QR flow. QR flow is the generalization of non-periodic Toda flow. The connection between Toda flow and QR algorithm was observed by Symes in [Sym82]. Let f is an analytic function in an open domain containing the spectrum of X_0. Then we can define

$$B(X) = f_l(X) - f_u(X), \tag{2.17}$$

where the subscripts l and u denote the lower and upper triangular parts of $f(X)$, respectively. Let us assume X is symmetric and $Q\Lambda Q^T$ is its factorization, here Q is an orthogonal matrix and Λ is a diagonal matrix. So we can say that $f(X) = Qf(\Lambda)Q^T$ is also symmetric, and hence, $B(X)$ is skew symmetric. QR flow is originated from the fact that for a symmetric positive definite initial condition X_0 there exists a connection between the isospectral flow (2.1) and the iterates of the familiar QR method for finding the eigenvalues of a matrix. QR method consists in finding the decomposition

$$X_k = Q_k U_k \tag{2.18}$$

where Q_k is an orthogonal matrix and U_k is an upper triangular matrix, with positive diagonal entries (Golub and van Loan 1989), $k = 1, 2, \cdots n$. The next matrix in the sequence is then obtained as

$$X_{k+1} = U_k Q_k \tag{2.19}$$

and, when X_0 is symmetric and positive definite, the method converges at an exponential speed to a diagonal matrix whose entries are the eigenvalues of X_0. It is easy to verify that

$$X_{k+1} = Q_k^T X_k Q_k, \tag{2.20}$$

rendering the connection with the isospectral flow is less obscure. If X_0 is symmetric and positive definite, choosing $f(x) = \log x$, the matrices $X(1), X(2), ...,$ interpolate the iterates $X_1, X_2, ...,$ obtained this time with QR method for eigenvalues. As we said earlier this connection was discovered by [Sym82] and then later investegated by [Nan82, DNT83, DRTW91, Chu84, CD89, Lag91] and many others.

Other well known examples are *Eigenvalue problems* and *inverse eigenvalue problems for symmetric Toeplitz matrices* [FNO87].

There are existing numerical methods that are most commonly used to solve ODEs. Clearly, it is possible that we discretise the isospectral flows by traditional numerical methods (e.g. Runge–Kutta and multistep), but these methods cannot respect the isospectrality of the system; once $n \geq 3$, the numerical solution changes the eigenvalues [CIZ99]. We need to solve these flows by a method that respects it, since isospectrality is essential for applications ranging from classical mechanics, like Toda flows and N-body systems, to linear algebra, like QR flows and inverse eigenvalue problems [CIZ97, CIZ99, Zan98]. Also, the detailed analysis done in [Zan98] shows the negative results for classical ODE methods.

Now, while we talk that the behaviour of the classical ODE methods in the preservation of conservation laws is generally poor, the question that comes to our mind is: Is it possible to devise numerical methods that are isospectral? The answer is positive. Consider again the lax pair (2.1)

$$X^{'} = [B(t, X), X], \quad t \geq 0,$$

with X_0, a symmetric initial condition and $B(X)$, a skew-symmetric matrix function. If we solve the system (2.1) with standard ODE method, we are bound to lose isospectrality. Recalling the proof of isospectrality of the system (2.1) from above, we see that $Q(t)$ is the solution of

$$Q^{'} = B(t, X)Q, \quad t \geq 0, \quad Q(0) = I,$$

which is an orthogonal flow because $B(X)$ is skew-symmetric and $X(t)$ can be obtained by similarity transformation,

$$X(t) = Q(t)X_0Q(t)^T.$$

Hence, the main idea of the isospectral methods is to solve the orthogonal flow instead and then approximate $X(t)$. As we know that the majority of numerical methods for ODEs do not preserve quadratic conservation laws, an easy way to construct orthogonal methods is by means of projection methods [LD94], evaluation of QR factorization can be seen in [GVL89] and [Dem90]. There are other examples of Runge–Kutta schemes [CIZ97][CIZ97] and [DRVV94], Cayley transforms [GVL89] and [DVV99], Modified Gauss–Legendre schemes, Semi-explicit methods [CIZ97] and the Lie group methods; Magnus expansion and Fer expansion. We discuss Lie group and Lie algebra in more detail in the following section.

2.2 Lie groups and Lie algebra

In this section we assemble the elements of Lie groups and Lie algebras. We briefly state background theory and introduce differentiable manifolds, Lie groups and their properties referring mainly to [IMKNZ99].
Lie groups and Lie algebras have originated by Sophus Lie (1842-1899) while solving differential equations by quadrature, using symmetry methods. In the twentieth century an abstract view of Lie group theory emerged. This formulation simplifies mathematical analysis and has become popular in Mathematics. However, the abstract theory concentrates on understanding mathematical structures rather than exposing applications in solving differential equations. Therefore, it is not clearly known to most applied mathematicians that Lie groups are really very useful in applied and computational mathematics. Traditionally, numerical integration of ordinary differential equations(ODEs) is basically the concept of solving the initial value problems of the form

$$x^{'} = f(t, x), \ t \geq 0, \ x(0) = x_0, \ x(t) \in \mathbb{R}^n \tag{2.21}$$

where f is vector field on $\mathbb{R}^n \times \mathbb{R}^n$. Traditional methods for numerical integration, like Euler forward method, Runge-Kutta and multistep methods iterate step by step by adding a vector on each iteration,

$$x_{n+1} = x_n + ha_n, \tag{2.22}$$

where a_n is calculated by the numerical method which we use and h is the stepsize. We can say that *classical integrators are formulated using a set of basic notations given by translations on $\mathbb{R}^n$ to advance the numerical solution.* A major motivation for Lie group methods is the possibility of replacing the domain $\mathbb{R}^n$ with more general configuration spaces and replacing translation on $\mathbb{R}^n$ by more general families of basic motions on the domain. To use Lie group methods and Lie algebra we should have an abstract view of differential manifolds, which is the domain on which differential equations evolve.

Definition 2.2.1. A d-dimensional manifold $\mathcal{M}$ is a d-dimensional smooth surface $\mathcal{M} \subset \mathbb{R}^n$ for some $n \geq d$.

Definition 2.2.2. Let $\mathcal{M}$ be a d-dimensional manifold and suppose that $\rho(t) \in \mathcal{M}$ is a smooth curve such that $\rho(0) = p$. A tangent vector at p is defined as

$$\mathbf{a} = \frac{d\rho(t)}{dt} \, |_{t=0} \, .$$

The set of all tangents at p is called the tangent space at p and denoted by $T\mathcal{M}|_p$. It has the structure of a d-dimensional linear space: if $\mathbf{a}$, $\mathbf{b} \in T\mathcal{M}|_p$ then $\mathbf{a} + \mathbf{b} \in T\mathcal{M}|_p$ and $\alpha\mathbf{a} \in T\mathcal{M}|_p$ for any real α. The collection of all tangent spaces at all points $p \in \mathcal{M}$ is called the tangent bundle of $\mathcal{M}$ and denoted by $T\mathcal{M} = \bigcup_{p \in \mathcal{M}} T\mathcal{M}|_p$.

Definition 2.2.3. A (tangent) vector field on $\mathcal{M}$ is a smooth function $F : \mathcal{M} \to T\mathcal{M}$ such that $F(p) \in T\mathcal{M}|_p$ for all $p \in \mathcal{M}$. The collection of all vector fields on $\mathcal{M}$ is denoted by $\mathfrak{X}(\mathcal{M})$.

Addition and scalar multiplication of vector fields are defined pointwise in a natural way as $(F + G)(p) = F(p) + G(p)$ and $(\alpha F)(p) = \alpha(F(p))$. If $F, G \in \mathfrak{X}(\mathcal{M})$ then also $F + G \in \mathfrak{X}(\mathcal{M})$ and $\alpha F \in \mathfrak{X}(\mathcal{M})$ for all real α.

Definition 2.2.4. Let F be a tangent vector field on $\mathcal{M}$. By a differential equation (evolving) on $\mathcal{M}$ we mean a differential equation of the form

$$\mathbf{y}' = F(\mathbf{y}), \qquad t \geq 0, \qquad \mathbf{y}(0) \in \mathcal{M}, \tag{2.23}$$

where $F \in \mathfrak{X}(\mathcal{M})$. Whenever convenient, we allow F in (2.23) to be a function of time, $F = F(t, \mathbf{y})$. The flow of F is the solution operator $\Psi_{t,F} : \mathcal{M} \to \mathcal{M}$ such that

$$\mathbf{y}(t) = \Psi_{t,F}(\mathbf{y}(0))$$

solves (2.23).

Lemma 2.2.5. *Given two vector fields F, G on $\mathbb{R}^n$, the commutator $H = [F, G]$ can be computed component wise at a given point $\boldsymbol{y} \in \mathbb{R}^n$ as*

$$H_i(\boldsymbol{y}) = \sum_{j=1}^{n} \Big\{ G_j(\boldsymbol{y}) \frac{\partial F_i(\boldsymbol{y})}{\partial y_j} - F_j(\boldsymbol{y}) \frac{\partial G_i(\boldsymbol{y})}{\partial y_j} \Big\}. \tag{2.24}$$

Lemma 2.2.6. *If $F, G \in \mathfrak{X}(\mathcal{M})$ then*

$$H = [F, G] \in \mathfrak{X}(\mathcal{M}).$$

Lemma 2.2.7. *Two flows $\Psi_{s,F}$ and $\Psi_{t,G}$ commute if and only if*

$$[F, G] = 0.$$

From (2.24) one may prove the following important features of the commutator which should be familiar in the special case (which we will encounter soon again) of a commutator of two matrices.

Definition 2.2.8. A Lie algebra is a linear space V equipped with a Lie bracket, a bilinear, skew-symmetric mapping

$$[\cdot, \cdot] : \; V \times V \to g$$

that obeys identities ((2.25) − (2.28)) from Lemma 2.2.9.

Lemma 2.2.9. *The commutator of vector fields satisfies the identities*

$$[F, G] = -[G, F], \qquad (\textit{skew symmetry}), \tag{2.25}$$

$$[\alpha F, G] = \alpha [F, G], \qquad \textit{for } \alpha \in \mathbb{R}, \tag{2.26}$$

$$[F + G, H] = [F, H] + [G, H], \qquad (\textit{bilinearity}), \tag{2.27}$$

$$0 = [F, [G, H]] + [G, [H, F]] + [H, [F, G]], \quad (\textit{Jacobi's identity}). \tag{2.28}$$

Definition 2.2.10. A Lie algebra of vector fields is a collection of vector fields which is closed under linear combination and commutation. In other words, letting $\mathfrak{g}$ denote the Lie algebra,

$$B \in \mathfrak{g} \quad \Rightarrow \quad \alpha B \in \mathfrak{g} \text{ for all } \alpha \in \mathbb{R}.$$

$$B_1, B_2 \in \mathfrak{g} \quad \Rightarrow \quad B_1 + B_2, \; [B_1, B_2] \in \mathfrak{g}.$$

Given a collection of vector fields $\mathbf{B} = \{B_1, B_2, \ldots\}$, the least Lie algebra of vector fields containing $\mathbf{B}$ is called the Lie algebra generated by $\mathbf{B}$.

Definition 2.2.11. A Lie algebra homomorphism is a linear map between two Lie algebras, $\varphi: \mathfrak{g} \times \mathfrak{h}$, satisfying the identity

$$\varphi([v,w]_{\mathfrak{g}}) = [\varphi(v), \varphi(w)]_{\mathfrak{h}}, \qquad v, w \in \mathfrak{g}.$$

An invertible homomorphism is called an isomorphism.

Definition 2.2.12. A Lie group is a differential manifold $\mathcal{G}$ equipped with a product $\cdot: \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ satisfying

$$\begin{aligned}
&p \cdot (q \cdot r) = (p \cdot q) \cdot r \ \forall p, q, r \in \mathcal{G} && \text{(associativity)},\\
&\exists \mathbf{I} \in \mathcal{G} \text{ such that } \mathbf{I} \cdot p = p \cdot \mathbf{I} = p \ \forall p \in \mathcal{G} && \text{(identity element)},\\
&\forall p \in \mathcal{G} \ \exists p^{-1} \in \mathcal{G} \text{ such that } p^{-1} \cdot p = \mathbf{I} && \text{(inverse)},\\
&\text{The maps } (p,r) \mapsto p \cdot r \text{ and } p \mapsto p^{-1} \text{ are smooth functions} && \text{(smoothness)}.
\end{aligned}$$

Definition 2.2.13. An action of a Lie group $\mathcal{G}$ on a manifold $\mathcal{M}$ is a smooth map $\Lambda: \mathcal{G} \times \mathcal{M} \to \mathcal{M}$ satisfying

$$\begin{aligned}
\Lambda(\mathbf{I}, \mathbf{y}) &= \mathbf{y} \quad \forall \mathbf{y} \in \mathcal{M}.\\
\Lambda(p, \Lambda(r, \mathbf{y})) &= \Lambda(p \cdot r, \mathbf{y}) \ \ \forall p, r \in \mathcal{G}, \ \forall \mathbf{y} \in \mathcal{M}.
\end{aligned} \tag{2.29}$$

If this relation does hold only in a local sense, for all elements p and r sufficiently close to the identity $\mathbf{I} \in \mathcal{G}$, we say that Λ is local action.

Definition 2.2.14. Let $\mathcal{G}$ be a Lie group and $\mathfrak{g}$ its Lie algebra. The exponential mapping $\exp: \mathfrak{g} \to \mathcal{G}$ is defined as $\exp(a) = \sigma(1)$ where $\sigma(t) \in \mathcal{G}$ satisfies the differential equation

$$\sigma'(t) = a\sigma(t), \qquad \sigma(0) = \mathbf{I}.$$

Definition 2.2.15. Let $p \in \mathcal{G}$ and let $\sigma(t)$ be a smooth curve on $\mathcal{G}$ such that $\sigma(0) = \mathbf{I}$ and $\sigma'(0) = b \in \mathfrak{g}$. The adjoint representation is defined as

$$\mathrm{Ad}_p(b) = \frac{d}{dt} p\sigma(t)p^{-1}|_{t=0}. \tag{2.30}$$

The derivative of Ad with respect to the first argument is denoted ad. Let $\rho(s)$ be a smooth curve on $\mathcal{G}$ such that $\rho(0) = \mathbf{I}$ and $\rho'(0) = a$. Now we know that:

$$\mathrm{Ad}_a(b) = \frac{d}{ds} Ad_{\rho(s)}(b)|_{s=0} = [a, b]. \tag{2.31}$$

The following formulae show that Ad is both a linear group action (of $\mathcal{G}$ on $\mathfrak{g}$) and also that for a fixed argument p it is a Lie-algebra isomorphism of $\mathfrak{g}$ onto itself:

$$\mathrm{Ad}_p(a) \in \mathfrak{g} \qquad \text{for all } p \in \mathcal{G},\ a \in \mathfrak{g} \tag{2.32}$$

$$\mathrm{Ad}_p \circ \mathrm{Ad}_q = \mathrm{Ad}_{pq} \tag{2.33}$$

$$\mathrm{Ad}_p(a+b) = \mathrm{Ad}_p(a) + \mathrm{Ad}_p(b) \tag{2.34}$$

$$\mathrm{Ad}_p([a,b]) = [\mathrm{Ad}_p(a), \mathrm{Ad}_p(b)]. \tag{2.35}$$

Note that according to (2.34) both Ad_p and ad_a are linear in their second argument, hence they may be regarded as matrices acting on the linear space $\mathfrak{g}$. This gives meaning to the following important formula relating Ad, ad and the exponential mapping:

$$\mathrm{Ad}_{\exp(a)} = \mathrm{expm}(\mathrm{ad}_a). \tag{2.36}$$

Definition 2.2.16. A real matrix Lie group is a smooth subset $\mathcal{G} \subseteq \mathbb{R}^{n\times n}$, closed under matrix products and matrix inversion. We let $\mathbf{I} \in \mathcal{G}$ denote the identity matrix.

Definition 2.2.17. The Lie algebra $\mathfrak{g}$ of a matrix Lie group $\mathcal{G}$ is the linear subspace $\mathfrak{g} \subseteq \mathbb{R}^{n\times n}$ consisting of all matrices of the form

$$\mathfrak{g} = \left\{ A \in \mathbb{R}^{n\times n} :\ A = \frac{d\rho(s)}{ds}\,|_{s=0} \right\},$$

where $\rho(s) \in \mathcal{G}$ is a smooth curve such that $\rho(0) = \mathbf{I}$. The space $\mathfrak{g}$ is closed under matrix additions, scalar multiplication and the matrix commutator

$$[A,B] = AB - BA. \tag{2.37}$$

Complex matrix Lie groups and algebras are defined similarly.

Definition 2.2.18. A differential equation on a matrix Lie group is an equation of the form

$$Y' = A(t,Y)Y, \qquad t \geq 0, \qquad Y(0) \in \mathcal{G}, \tag{2.38}$$

where $A:\ \mathbb{R} \times \mathcal{G} \to \mathfrak{g}$ and AY is the usual matrix product between $A \in \mathfrak{g}$ and $Y \in \mathcal{G}$.

This can be verified that this is the special case of the general form of a differential equation on a manifold, where $\mathcal{M} = \mathcal{G}$ is a matrix Lie group and the action Λ is taken to be the left (matrix) multiplication in $\mathcal{G}$,

$$\Lambda(R,Y) = RY.$$

We find

$$\lambda_*(A)Y = AY.$$

Since $\mathfrak{g}$ is defined as the collection of all tangent directions at $\mathbf{I} \in \mathcal{G}$ and matrix multiplication by Y is an invertible mapping, we see that any tangent at Y can be written in the form AY and all differential equations on $\mathcal{G}$ can be written in the form (2.38).

Definition 2.2.19. The exponential mapping expm : $\mathfrak{g} \to \mathcal{G}$ is defined as

$$\mathrm{expm}(A) = \sum_{j=0}^{\infty} \frac{A_j}{j!}. \tag{2.39}$$

The adjoint representation, Ad, and its derivative, *ad*, are defined as

$$\mathrm{Ad}_P(A) = PAP^{-1} \tag{2.40}$$

$$\mathrm{ad}_A(B) = AB - BA = [A, B]. \tag{2.41}$$

2.3 Some examples of Lie groups and algebras

We introduce here some concrete examples of Lie groups and algebras. In each case it is easy to verify that all the axioms of a group or an algebra, as the case might be, are fulfilled.

- The set of all real $n \times n$ nonsingular matrices is a (multiplicative) Lie group, the general linear group $GL(n)$. The corresponding Lie algebra is the set $\mathbb{R}^{n\times n}$ of all $n \times n$ real matrices which, we denote by $\mathfrak{gl}(N)$.

  The general linear group and algebra can be defined over other fields than $\mathbb{R}$ in which case we communicate this in the second argument. For example, $GL(n, \mathbb{C})$ consists of all nonsingular $n \times n$ complex matrices.

- All members of $GL(n)$ with unit determinant form the special linear group $SL(n)$. Its Lie algebra, $\mathfrak{sl}(N)$, consists of all matrices in $\mathfrak{gl}(N)$ with zero trace.

- All the matrices $X \in SL(4)$ such that $XJX^T = J$, where

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

  form the Lorenz group $SO(3,1)$. Its Lie algebra $\mathfrak{so}(3,1)$ is made out of all $F \in \mathfrak{gl}(4)$ such that $FJ + JF^T = \mathbf{O}$.

- $N \times N$ real orthogonal matrices form the *orthogonal group* $O(n)$, whose Lie algebra $\mathfrak{so}(N)$ consists of $N \times N$ skew-symmetric matrices.

The set $SO(n) = SL(N) \cap O(N)$, consisting of $N \times N$ real orthogonal matrices with a unit determinant, is the *special orthogonal group*. Its Lie algebra is $\mathfrak{so}(N)$. This is not contradictory; this is to be noted, we never claimed that two different Lie groups must have different Lie algebras. If $\mathcal{G}$ is a Lie group and $\mathcal{G}_{Id}$ its connected component such that $I \in \mathcal{G}_{Id}$ (precisely the situation with $O(n)$ and $SO(n)$, respectively) then they produce the same Lie algebra.

- The set of all $(2N) \times (2N)$ real matrices X such that $XJX^T = J$, where

$$J = \begin{bmatrix} \mathbf{O}_N & \mathbf{I}_N \\ -\mathbf{I}_N & \mathbf{O}_N \end{bmatrix},$$

  is the *symplectic group* and is denoted by $Sp(N)$. (The Jacobian of the flow of a Hamiltonian ODE system evolves in $Sp(N)$.) The corresponding Lie algebra, $\mathfrak{sp}(N)$, consists of $F \in \mathfrak{gl}(2N)$ such that $FJ + JF^T = \mathbf{O}$.

- As an example of complex Lie groups, we mention the *unitary group* $U(N;\mathbb{C})$ of all the $N \times N$ complex unitary matrices: $X \in U(N;\mathbb{C})$ if and only if $XX^H = \mathbf{I}$. The Lie algebra corresponding to the $U(N;\mathbb{C})$ is the set $\mathfrak{u}(N;\mathbb{C})$ of all the skew-Hermitian matrices in $\mathfrak{gl}(N;\mathbb{C})$.

- Similarly, for the case of $O(n)$ and $SO(n)$, we obtain the *special unitary group* intersecting $U(N;\mathbb{C})$ with $SL(N;\mathbb{C})$. Its Lie algebra, $\mathfrak{su}(N;\mathbb{C})$, is composed of $N \times N$ complex skew-Hermitian and traceless matrices

In this dissertation we use Magnus expansion, which is a Lie group expansion, to discretise Bloch–Iserles equations and generalised double bracket flow. Let us have a brief look at the Magnus expansion in the following section.

2.4 Magnus expansion and binary rooted trees

In this section we explain the Magnus expansion [Mag54] which we apply in the next chapters to discretise Bloch–Iserles equations and generalised double bracket flow. The main references for this section are [Mag54], [Ise99] and [IN99] other references if any, are mentioned explicitly.

Magnus expansion provides an exponential representation of the solution of linear ordinary differential equation. Consider a matrix differential equation

$$Y'(t) = A(t)Y(t), \quad t \geq 0, \quad Y(0) = Y_0, \tag{2.42}$$

where A is a $n \times n$ matrix.
For $n = 1$, solution of (2.42) is

$$Y(t) = e^{\int_0^t A(\xi)d\xi} Y_0, \quad t \geq 0. \tag{2.43}$$

For $n > 1$ and if A is not a constant matrix then the above expression (2.43) is no longer the solution of the problem.

Magnus [Mag54] proposed the solution to the matrix initial value problem and he expressed the solution in the form

$$Y(t) = e^{\Omega} Y_0. \tag{2.44}$$

The expression for Ω is (by Felix Hausdorff [Hau06]) given by

$$\Omega' = \sum_{m=0}^{\infty} \frac{B_m}{m!} \mathrm{ad}_\Omega^m A, \quad t \geq 0, \quad \Omega(0) = 0, \tag{2.45}$$

where $B_m, m \in \mathbb{Z}$ are Bernoulli's numbers and ad_Ω^m is an iterated commutator and we have

$$\mathrm{ad}_\Omega^0 A = A, \quad \mathrm{ad}_\Omega^1 A = [\Omega, A], \quad \mathrm{ad}_\Omega^2 A = [\Omega, [\Omega, A]], ..., \quad \mathrm{ad}_\Omega^m A = [\Omega, \mathrm{ad}_\Omega^{m-1} A],$$

where $[\Omega, A] = \Omega A - A\Omega$.

Magnus observed in [Mag54] that Ω can be written as a linear combination of multiple integrals. Employing Picard's iteration,

$$\Omega_0(t) = 0,$$

$$\Omega_{s+1}(t) = \int_0^t \sum_{m=0}^{\infty} \frac{B_m}{m!} ad_{\Omega_s(\xi)}^m A(\xi).$$

it is possible to show that

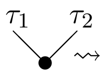
$$\begin{aligned} \Omega(t) \quad &= \int_0^t A(\xi)d\xi - \frac{1}{2}\int_0^t \int_0^{\xi_1} [A(\xi_2), A(\xi_1)] d\xi_2 d\xi_1 \\ &+ \frac{1}{12}\int_0^t \int_0^{\xi_1} \int_0^{\xi_1} [A(\xi_3), [A(\xi_2), A(\xi_1)]] d\xi_3 d\xi_2 d\xi_1 \\ &+ \frac{1}{4}\int_0^t \int_0^{\xi_1} \int_0^{\xi_1} [[A(\xi_3), A(\xi_2)], A(\xi_1)] d\xi_3 d\xi_2 d\xi_1 + \cdots . \end{aligned}$$

Calculating further terms get out of hand because it is growing exponentially with fourfold integrals and three nested commutators. Therefore, Iserles and Nørsett [IN99] proposed an alternative way to use binary rooted trees as a shorthand for expansion terms. Before going to the tree representation of the Magnus expansion, let us have a brief view of basic concepts of graph theory [Har69].

- Let $V = \{v_1, v_2, ..., v_r\}$ be a finite set of distinct vertices and $E = V \times V$ a set of edges. Then $G = \langle V, E \rangle$ is a *graph*.
- The graph is said to be *connected* if there is a path between any two vertices.
- It is a *tree* if exactly one path links every two vertices.
- The ordered set $\{(v_{s_l}, v_{t_l}) : l = 1, 2, ..., r\}$ of edges is a *path* from $v_i \in V$ to $v_j \in V, i \neq j$, if $s_1 = i, t_l = s_{l+1}, l = 1, 2, ..., r-1$ and $t_r = j$.
- The pair $T = (G, w)$, where G is a tree and $w \in V$ is its root, is called a *rooted tree*. There exist a natural partial order on T: we say that $v_i \prec v_j$ if v_i precedes v_j in the unique path extending from the root w to v_j. In that case v_i is the ancestor of v_j, while v_j is the successor of v_i.
- If $v_i \prec v_j$ and there is no $v_k \in V$ such that $v_i \prec v_k \prec v_j$, we say that v_i is the parent of v_j and v_j is the child of v_i. Childless vertices are called leaves.
- If each vertex in a rooted tree has at most two children, T is called a *binary tree*. If each vertex has either exactly two children or is a leaf, T is said to be a *strictly binary tree*.

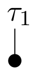
Constructing the binary rooted trees for the expansion [Ise99], we commence by assigning to A a trivial tree

$$\bullet \rightsquigarrow A,$$



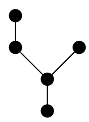
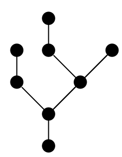
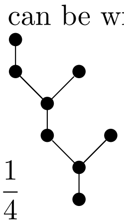
$$\rightsquigarrow [H_{\tau_1}, H_{\tau_2}],$$

and



$$\rightsquigarrow \int_0^t H_{\tau_1}(\xi) d\xi.$$

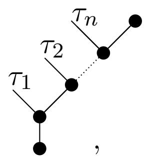
A is an integrable $n \times n$ matrix function, we define a map $\tau \rightarrow H_\tau$ from $\mathbb{T}$, a subset of binary rooted trees into $n \times n$ matrix functions. Thus, Ω can be written as

$\Omega =$  $- \frac{1}{2}$  $+ \frac{1}{12}$  $+ \frac{1}{4}$  $+ \cdots .$

Following the tree algorithm proposed in [Ise99], writing the Magnus expansion in the form

$$\Omega(t) = \sum_{r=1}^{\infty} \sum_{\tau \in \mathbb{T}_r} \alpha(\tau) H_\tau(t),$$

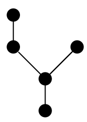
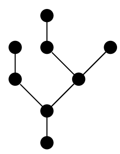
where the trees τs are of the form

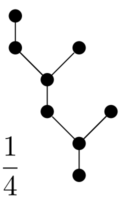
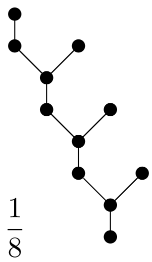
$\tau =$  , (2.46)

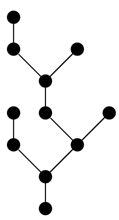
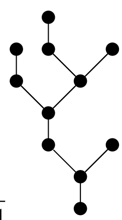
and the constant $\alpha(\tau)$ is given by

$$\alpha(\tau) = \frac{\mathrm{B}_s}{s!} \prod_{i=1}^{s} \alpha(\tau_i), \quad s \in \mathbb{N},$$

where B_s is the sth Bernoulli number and the trees $\tau_1, \tau_2, \cdots$ are featured earlier in the expansion. We construct the next level trees,

$\Omega =$  $-\frac{1}{2}$  $+\frac{1}{12}$  (2.47)

$+\frac{1}{4}$  $-\frac{1}{8}$ 

$-\frac{1}{24}$  $-\frac{1}{24}$  $\cdots$.

The Magnus expansion absolutely converges for every $t \geq 0$ such that

$$\int_0^t A(\xi) d\xi \leq \int_0^{2\pi} \frac{d\xi}{4 + \xi[1 - \cot(\frac{\xi}{2})]} \approx 1.086868702.$$

In the next chapter we discretise the Bloch–Iserles equations using Magnus expansion that helps preserving the isospectrality of the solution space and gives the desired structure underlying our approach. Moreover, the shorthand of binary rooted trees helps to reduce the computational cost. We see that the solution terms in the Taylor expansion using the Mangus series, grow exponentially and manual computation becomes very expensive. Our approach of using binary rooted trees makes it possible to formulate the explicit representation of the solution.

Chapter 3

Bloch–Iserles equations

This chapter is based on my paper "On solving an isospectral flow" [Kau16]. In this chapter we expand the solution of the matrix ordinary differential system, namely Bloch–Iserles equations, of the form $X' = [N, X^2]$, $\ t \geq 0$, $\ X(0) = X_0 \in \mathrm{Sym}(n)$, $\ N \in \mathfrak{so}(n)$, where $\mathrm{Sym}(n)$ denotes the space of real $n \times n$ symmetric matrices and $\mathfrak{so}(n)$ denotes the Lie algebra of real $n \times n$ skew-symmetric matrices. The flow is solved using explicit Magnus expansion, which respects the isospectrality of the system. We represent the terms of expansion as binary rooted trees and deduce an explicit formalism to construct the trees recursively.

3.1 Introduction

We are concerned with the discretization of the matrix differential equation

$$X' = [N, X^2], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n), \quad N \in \mathfrak{so}(n). \tag{3.1}$$

The system (3.1) is known as the Bloch–Iserles (BI) equations. It is isospectral (preserves the eigenvalues of $X(t)$), is endowed with a Poisson structure and is integrable as proved by Bloch and Iserles [BI06]. We discretise this system using a similar approach, for instance in [Ise02] and [Cas04]. However, solving BI is much more complicated since it contains X^2 in the expression.

The above system is of interest for a number of reasons. Firstly, we can easily verify that it can be written in the form

$$X' = [N, X]X + X[N, X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n). \tag{3.2}$$

For $N \in \mathfrak{so}(n)$ and $X \in \mathrm{Sym}(n)$, we have $[N, X] \in \ \mathrm{Sym}(n)$, therefore it is a special case of a *congruent flow*

$$X^{'} = A(X)X + XA^{T}(X), \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n), \tag{3.3}$$

where $A : \mathrm{Sym}(n) \to M(n)$, where $M(n)$ is the set of real $n \times n$ matrices, is sufficiently smooth. It is easy to verify that $X(t) = V(t)X_0V^T(t)$, where $V^{'} = A(VX_0V^T)V,\ V(0) = I$. That means the solution is an outcome of the *general linear group* $\mathrm{GL}(n)$ acting on $\mathrm{Sym}(n)$ by congruence. That proves that the *signature* of $X(t)$ is constant [HJ91]. Another interesting aspect of the given set of equations is that they are dual to the *generalized rigid body equations*

$$M^{'} = [\Omega, M], \quad t \geq 0, \quad M(0) \in \mathfrak{so}(n),$$

where $M = \Omega J + J\Omega,\ J \in \mathrm{Sym}(n)$ therefore $\Omega \in \mathfrak{so}(n)$ [Man76].

Also, it is clear that (3.1) can be rewritten in the form

$$X' = [XN + NX, X], \quad t \geq 0, \quad X(0) = X_0 \in \ \mathrm{Sym}(n), \quad N \in \mathfrak{so}(n).$$

Since $XN + NX \in \mathfrak{so}(n)$ for $X \in\ \mathrm{Sym}(n), \quad N \in \mathfrak{so}(n)$, it follows that the system (3.1) is indeed isospectral.

There are traditional numerical methods (e.g. Runge–Kutta and multistep) to discretise (3.1), but, as we discussed earlier, this is unfortunate that once $n \geq 3$, these methods cannot respect the isospectrality of the system and hence we are bound to lose the structural feature of the system, i.e. the numerical solution changes the eigenvalues [CIZ99]. Isospectrality is essential for applications ranging from classical mechanics, like Toda flows and N-body systems, to linear algebra, like QR flows and inverse eigenvalue problems, so we need to solve (3.1) by a method that respects it [CIZ97, CIZ99, Zan98].

In this chapter we solve the given isospectral flow using the method of Magnus series. We represent the solution of (3.1) in the form $X(t) = e^{\Omega(t)}X_0e^{-\Omega(t)}$, where instead of computing X at the first place, we obtain the Taylor expansion of Ω and in each step an orthogonal matrix $Q(t) = e^{\Omega(t)}$ is evaluated. Approximating $X(t)$ by $X(t) = e^{\Omega(t)}X_0e^{-\Omega(t)}$, ensures automatically that the numerical solution, being similar to X_0, is isospectral. We will see that the Taylor expansion of Ω can be formed algorithmically from X_0 and N and linear combinations of their commutators and anti-commutators. Our purpose is to determine the rules for finding the terms of Ω to an arbitrary accuracy. For the solution, first we convert the isospectral flow to a Lie-group flow and then translate it into a Lie-algebraic equation. It is observed that this method preserves the isospectrality (Figure 3.2) and gives the desired structure of the solution with large time steps. In section 3.2 we solve the given system of differential equations using the Magnus expansion to obtain the Taylor expansion of Ω. We will see that the number of terms in each iteration grow exponentially therefore the complexity of manual computation becomes prohibitive. Finally, in section

3.3 we represent the terms by binary rooted trees and form an algorithm to obtain the recursive formula that is used to construct the next generation trees and to calculate the coefficient of each tree. This technique is the inauguration to the general setting, i.e. the explicit representation of the solution of (3.1) when $B(X)$ can be represented in a finite "alphabet", namely X_0 and N. Once we construct the trees, it is comparatively easier to translate them into terms containing X_0 and N. Also, it is remarkable that the skew-symmetry and Jacobi identity obeyed by the commutator help us to reduce the number of the terms by cancelling or writing certain terms as the linear combination of other terms.

3.2 An expansion of the solution

As stated above, the Bloch–Iserles system can be rewritten in the form

$$X^{'} = [B(X), X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n).$$

with $B(X) = NX + XN$, where $B(X) : \mathrm{Sym}(n) \to \mathfrak{so}(n)$. This system is seen to be isospectral and it is standard to verify that

$$X(t) = Q(t) X_0 Q^T(t), \quad t \geq 0, \tag{3.4}$$

where $Q(t) \in \mathrm{SO}(n)$ is the solution of

$$Q'(t) = (Q(t)X_0Q^T(t)N + NQ(t)X_0Q^T(t))Q(t), \quad Q(0) = I. \tag{3.5}$$

In a similar way as Magnus [Mag54] did for linear equations, our idea is to represent the solution of (3.5) in the form

$$Q(t) = e^{\Omega(t)},$$

where

$$\Omega' = \sum_{0}^{\infty} \frac{\mathrm{B}_r}{r!} \mathrm{ad}_{\Omega}^r (e^{\Omega} X_0 e^{-\Omega} N + N e^{\Omega} X_0 e^{-\Omega}), \quad \Omega(0) = 0. \tag{3.6}$$

Here $\mathrm{B}_m,\ m \in \mathbb{Z}$ are Bernoulli numbers and ad_{Ω}^r is an iterated commutator defined by

$$\mathrm{ad}_{\Omega}^0 A = A, \quad \mathrm{ad}_{\Omega}^1 A = [\Omega, A], \quad \mathrm{ad}_{\Omega}^2 A = [\Omega, [\Omega, A]], ..., \quad \mathrm{ad}_{\Omega}^m A = [\Omega, \mathrm{ad}_{\Omega}^{m-1} A],$$

where $[\Omega, A] = \Omega A - A\Omega$.

Now, taking $\Omega(t)=\sum_{m=0}^{\infty}\Omega_m t^m$ gives

$$\Omega'(t)=\sum_{m=0}^{\infty}(m+1)\Omega_{m+1}t^m \tag{3.7}$$

and this implies

$$\sum_{m=0}^{\infty}(m+1)\Omega_{m+1}t^m=\sum_{r=0}^{\infty}\frac{\mathrm{B}_r}{r!}\mathrm{ad}^r_{\Omega(t)}(e^{\Omega(t)}X_0e^{-\Omega(t)}N+Ne^{\Omega(t)}X_0e^{-\Omega(t)}). \tag{3.8}$$

Comparing coefficients of t^0,t^1,t^2... we get the values of $\Omega_1,\Omega_2,\Omega_3$... as follows

$$\begin{aligned}
\Omega_1 &= NX_0+X_0N,\\
\Omega_2 &= \frac{1}{2}(N[\Omega_1,X_0]+[\Omega_1,X_0]N),\\
\Omega_3 &= \frac{1}{3}(N[\Omega_2,X_0]+[\Omega_2,X_0]N)+\frac{1}{6}(N[\Omega_1,[\Omega_1,X_0]]+[\Omega_1,[\Omega_1,X_0]]N)\\
&\quad+\frac{1}{6}[\Omega_2,\Omega_1].
\end{aligned}$$

We denote $NX+XN=\{X\}$, thereby rewriting the above values of $\Omega_1,\Omega_2,\Omega_3\ldots.$ in a more succinct manner as

$$\begin{aligned}
\Omega_1 &= \{X_0\},\\
\Omega_2 &= \frac{1}{2}\{[\Omega_1,X_0]\},\\
\Omega_3 &= \frac{1}{3}\{[\Omega_2,X_0]\}+\frac{1}{6}\{[\Omega_1,[\Omega_1,X_0]]\}+\frac{1}{6}[\Omega_2,\Omega_1],\text{etc.}
\end{aligned} \tag{3.9}$$

Note that $X\in\mathrm{Sym}(n)$, $N\in\mathfrak{so}(n)$ implies that $\{X\}\in\mathfrak{so}(n)$. Introducing the curly bracket, i.e. featuring N implicitly in $NX+XN=\{X\}$ is important mainly for two reasons. Firstly, this helps to understand the recurrence of the terms in the expansion, else $XN+NX$ gets convoluted with other terms (either by getting cancelled or by adding up). Later in section 3.3, while defining the structure of binary rooted trees; it prevents from getting bicolored leaves, for instance, bicolored leaves in [Ise02].
Thus,

$$\Omega(t) = t\{X_0\}+\frac{1}{2}t^2\{[\{X_0\},X_0]\}$$

$$+t^3(\frac{1}{6}\{[\{[\{X_0\},X_0]\},X_0]\}+\frac{1}{6}\{[\{X_0\},[\{X_0\},X_0]]\}$$
$$+\frac{1}{12}[\{[\{X_0\},X_0]\},\{X_0\}])+\ldots. \tag{3.10}$$

In next term there are three nested commutators. These are getting increasingly complex with each iteration and it is clear that the number of such terms is growing exponentially. In this chapter we will represent these terms in terms of rooted trees, similar to the case of double-bracket flows in [Ise02]. This builds upon an idea of Iserles and Nørsett [IN99] to use binary rooted trees as a shorthand for expansion terms.

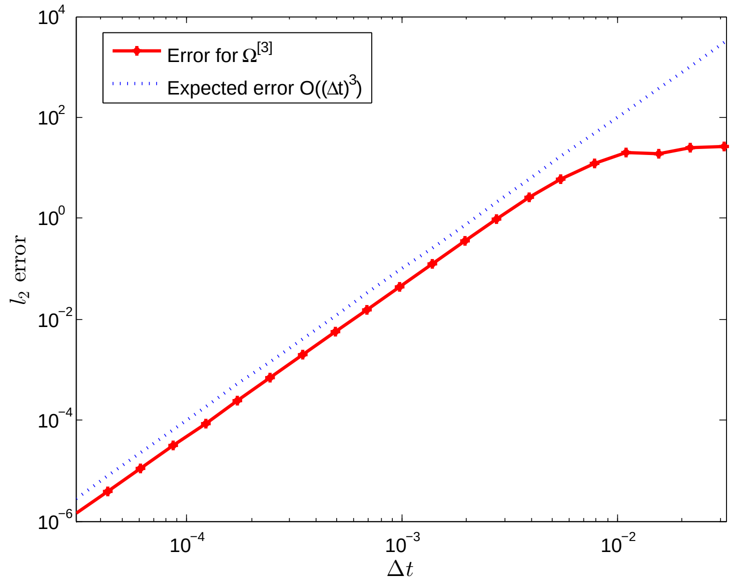


Figure 3.1: Global error on logarithmic scale across an interval [0,1] with different time steps, after truncating the Taylor expansion up to third order terms.

Before we follow the procedure to simplify the above expansion, preliminary error graph of this method and error graph for eigenvalues of this method, as compared to the MATLAB ode45 solver with built-in parameters, are presented in Figure 3.1 and Figure 3.2 respectively.

In Figure 3.1 we display error in the solution of (3.1) in the interval [0,1] for a range of different step sizes Δt. The error plot is generated by truncating the expansion up to order three and is compared against the theoretically expected error of $O((\Delta t)^3)$. The experiments were performed on random 25×25 matrices. Note that in Figure 3.1 we are interested in the case when $\Delta t \to 0$ (asymptotic limit), which corresponds to the left part

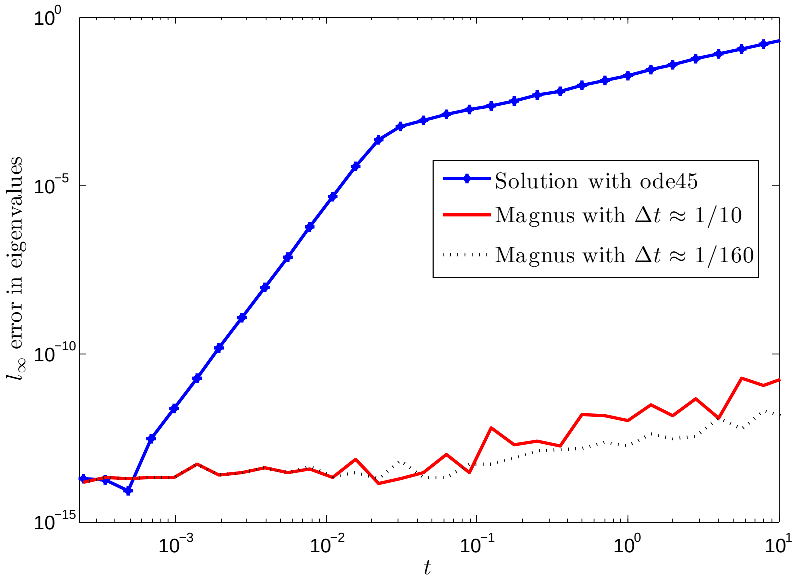


Figure 3.2: Error plot showing absolute error of eigenvalues of the two methods, Magnus expansion and ode45, on a logarithmic scale.

of the figure. In Figure 3.2 we calculate absolute error of eigenvalues of the two methods on a logarithmic scale. It is clearly seen that our method preserves the correct eigenvalues to machine accuracy (we note here that the machine epsilon 10^{-16} corresponds to relative error; however, we are calculating the absolute error in the graph). Despite a large time-step in the Magnus method, the error in eigenvalues stays very close to machine precision while the solution obtained using ode45 quickly strays away in terms of eigenvalues as time increases. This favourable behavior of the Magnus method is to be expected from the principles underlying our approach.

Also, we have computed numerically the solution of the system for random 3×3 matrices using Lie group method using Magnus expansion. In the Figure 3.3, the phase portraits $(X_{1,2}, X_{k,l})$ are displayed for $(k,l) = (1,1), (1,3), (2,2)$, $(2,3), (3,3)$, with random initial condition, here by $X_{k,l}$ we mean the kl^{th} element of the matrix X. It has been proved in [BI06] that the system is Lie-Poisson and in [BBI$^+$09] and [LT06] that it is integrable. So, it is clear that the behaviour of the solution displays the regularity and the solution curve lies on invariant tori. This is an indication of integrable Lie-Poisson structure. Similar behaviour is obtained for other randomly calculated matrices as well.

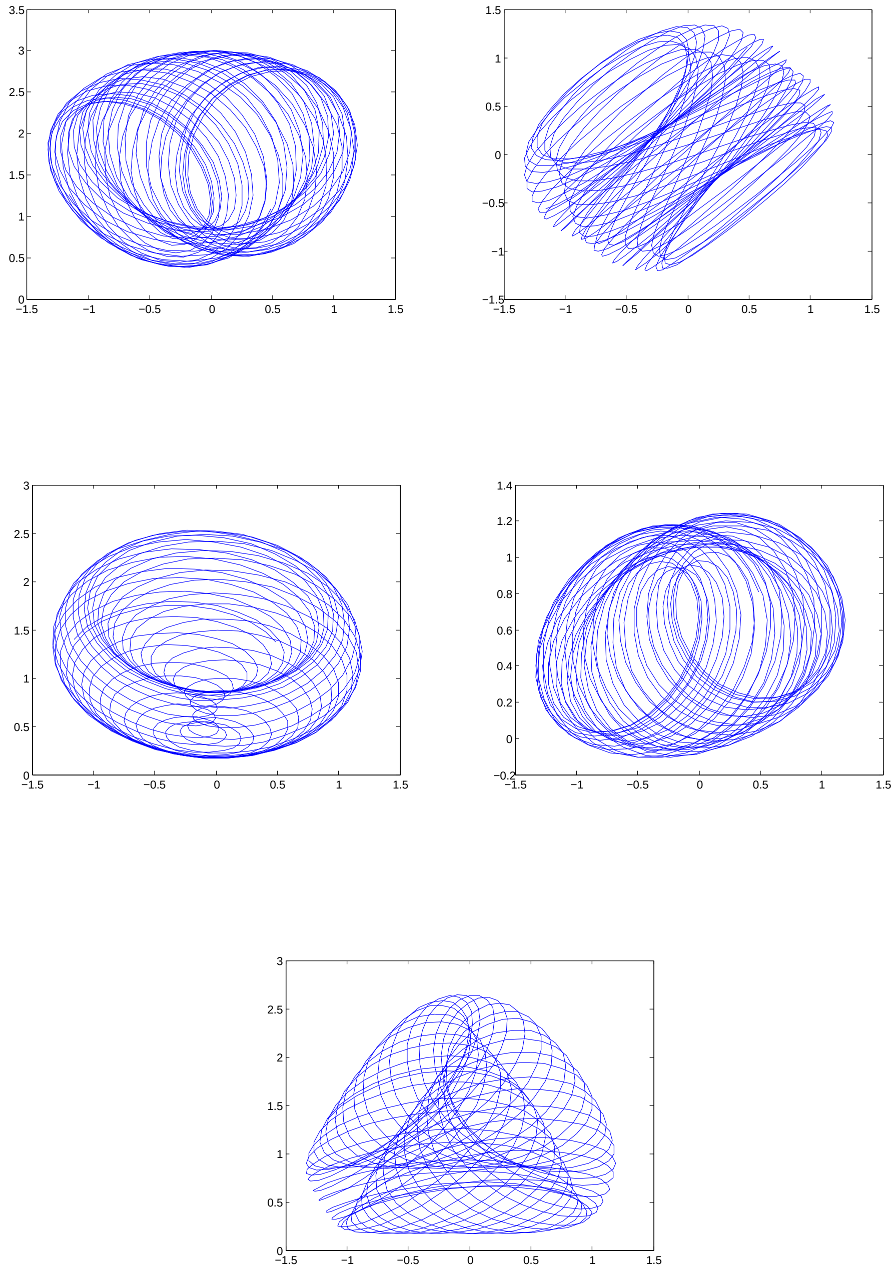


Figure 3.3: The phase portraits $(X_{1,2}, X_{k,l})$ for $(k, l) = (1, 1), (1, 3), (2, 2), (2, 3)$, $(3, 3)$, respectively for (3.1), with a random initial condition. Here, by $X_{k,l}$ we mean the kl^{th} element of the matrix X.

3.3 Representation by binary trees

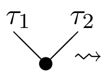
Once we look at the expansion Ω, we note that each term is written in just two 'letters' X_0 and N, hence belongs to a free structure generated by them: The matrix N features implicitly in the bracket $\{Z\} = NZ + ZN$. Thus we can say that each term is made of commutators and curly brackets with some expression Z, which itself has been formed from X_0 and N. We attempt to find the expansion of the solution in Taylor series of the form

$$\Omega(t) = \sum_{r=1}^{\infty} \sum_{\tau \in \mathbb{T}_r} \alpha(\tau) H_\tau \tag{3.11}$$

where $\mathbb{T}_r$ is the set of all binary trees of power r : A tree τ is of power $r \geq 1$ if r is the greatest integer such that $H_\tau = O(t^r)$, H_τ is an expression constructed from X_0 and N according to rules implicit in the structure of the tree τ which will be explained next, and α is a scalar coefficient. Using rooted trees as a shorthand for expansion terms is an approach introduced by [IN99], that leads to a framework that elucidates the structure of individual terms and their relationship. While constructing trees, we commence by assigning X_0 to a single node, i.e. a *trivial tree*,

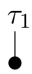
$$\bullet \rightsquigarrow X_0.$$

We define a function $\tau \to H_\tau$ from $\mathbb{T} = \bigcup_{r=1}^{\infty} \mathbb{T}_r$, a subset of binary rooted trees into $n \times n$ matrix functions by letting $H_\bullet = X_0$ and, by induction,



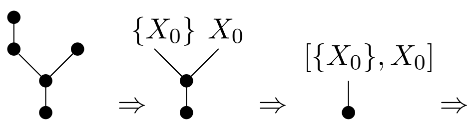
$\rightsquigarrow [H_{\tau_1}, H_{\tau_2}],$

and



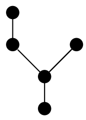
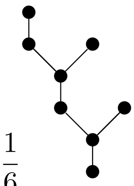
$\rightsquigarrow \{H_{\tau_1}\}.$

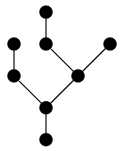
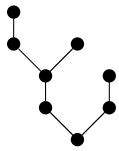
where H_{τ_1} and H_{τ_2} are already constructed expansion terms. For example,



$\Rightarrow \{[\{X_0\}, X_0]\} \Rightarrow \{[\Omega_1, X_0]\}.$

In particular (3.10) can be written as

$$\Omega \quad = \quad$$  $$t \; + \; \frac{1}{2}$$  $$t^2 \; + \frac{1}{6}$$  $$t^3 \qquad (3.12)$$

$$+\frac{1}{6}$$  $$t^3 + \frac{1}{12}$$  $$t^3 \; + \cdots .$$

To explore the general rules underlying this correspondence between trees and expansion terms, let us have again a look at the equation

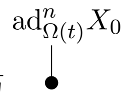
$$\sum_{m=0}^{\infty}(m+1)\Omega_{m+1}t^m = \sum_{r=0}^{\infty}\frac{\mathrm{B}_r}{r!}\mathrm{ad}^r_{\Omega(t)}\Big(e^{\Omega(t)}X_0e^{-\Omega(t)}N + Ne^{\Omega(t)}X_0e^{-\Omega(t)}\Big).$$

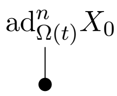
We know that

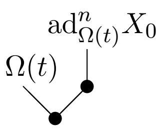
$$e^{\Omega}X_0e^{-\Omega} = \mathrm{Ad}_{\Omega}X_0 = e^{\mathrm{ad}_{\Omega}}X_0 = \sum_{n=0}^{\infty}\frac{1}{n!}\mathrm{ad}^n_{\Omega}X_0$$

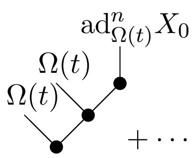
Thus

$$\sum_{m=0}^{\infty}(m+1)\Omega_{m+1}t^m \;=\; \sum_{r=0}^{\infty}\frac{\mathrm{B}_r}{r!}\mathrm{ad}^r_{\Omega(t)}\Big(N\sum_{n=0}^{\infty}\frac{1}{n!}\mathrm{ad}^n_{\Omega(t)}X_0 + \sum_{n=0}^{\infty}\frac{1}{n!}\mathrm{ad}^n_{\Omega(t)}X_0N\Big)$$

$$= \; \sum_{r=0}^{\infty}\frac{\mathrm{B}_r}{r!}\mathrm{ad}^r_{\Omega(t)}\sum_{n=0}^{\infty}\frac{1}{n!}$$ 

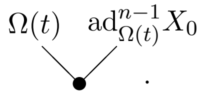
$$= \; \frac{\mathrm{B}_0}{0!}\sum_{n=0}^{\infty}\frac{1}{n!}$$ 

$$+\;\frac{\mathrm{B}_1}{1!}\sum_{n=0}^{\infty}\frac{1}{n!}$$ 

$$+\frac{\mathrm{B}_2}{2!}\sum_{n=0}^{\infty}\frac{1}{n!}$$ 

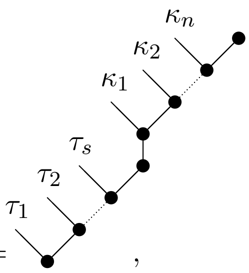
$$+\cdots . \qquad (3.13)$$

where

$\mathrm{ad}^{n}_{\Omega(t)} X_0 =$ 

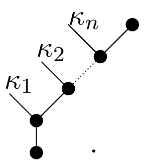
.

Clearly, in (3.13) we see that each tree here can be represented in the form

$T_{s,n} \ni \tau =$ 

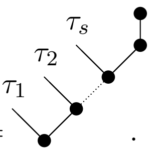
, (3.14)

where $s \in \{0, 1, 2, 3, \ldots\}$, $n \in \{0, 1, 2, 3, \ldots\}$. Here, the trees $\tau_1, \tau_2, \ldots \tau_s, \kappa_1, \kappa_2, \ldots \kappa_n$ have been featured earlier in the expansion, where $\tau_i \in \mathbb{T}_{p_i}$, and $\kappa_j \in \mathbb{T}_{q_j}$, $i = 1, 2, ..., s$, $j = 1, 2..., n$ and $p_1 + p_2 + ... + p_s + q_1 + q_2 + ... + q_n + 1 = r$. For $s = 0$ (for example, in the first four trees in (3.12)), the structure of the trees becomes

$T_{0,n} \ni \tau =$ 

. (3.15)

For $n = 0$ (for example, the fifth tree), this becomes

$T_{s,0} \ni \tau =$ 

. (3.16)

It is also possible to deduce the explicit form of the constant α by substituting the value of Ω from (3.11) in the form of $\alpha(\tau)$ and H_τ in (3.13) and simplifying it for $\alpha(\tau)$. Set

$\alpha\Big($  $\Big) = 1.$ (3.17)

Let $\tau \in \mathbb{T}_r, r \in \mathbb{N}$ and suppose that $\alpha(\tau_i)$ and $\alpha(\kappa_j)$ are known for $i = 1, 2, \ldots, s$, $j = 1, 2, \ldots, n$. Then

$$\alpha(\tau) = \frac{1}{r}\frac{\mathrm{B}_s}{s!}\frac{1}{n!}\prod_{i=1}^{s}\alpha(\tau_i)\prod_{j=i}^{n}\alpha(\kappa_j), \quad s, n \in \mathbb{N}, \tag{3.18}$$

where B_s is the sth Bernoulli number.

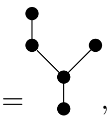
Now we have the general pattern for recursion. Suppose that $\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_{r-1}$ are known and also the coefficient α in these sets is known. To construct $\mathbb{T}_r$ we note that every τ is of the form (3.14) for some $s \in \{0, 1, 2, 3, \ldots, r-1\}$ and $n \in \{0, 1, 2, 3, \ldots, r-1\}$. For every such s and n we consider all the partitions $p_1 + p_2 + \ldots + p_s + q_1 + q_2 + \ldots + q_n + 1 = r$. For every partition we construct the tree τ in (3.14) and use (3.18) to determine the coefficient α. The trees which correspond to zero terms would be eliminated. Moreover, some trees can be replaced by linear combinations of other trees. Let us start from $\mathbb{T}_1$

(i) $\mathbb{T}_1$: For $s = 0, n = 0$, we have

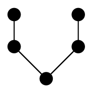
$\tau_1^1 =$  , $\quad \alpha(\tau_1^1) = \frac{1}{1} \cdot 1 \cdot \frac{1}{0!} = 1.$

(ii) $\mathbb{T}_2$:

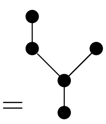
(1) For $s = 0, n = 1$, we can have only one possibility, $\kappa_1 =$ 

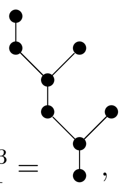
$\tau_1^2 =$  , $\quad \alpha(\tau_1^2) = \frac{1}{2} \cdot 1 \cdot \frac{1}{1!} = \frac{1}{2};$

(2) For $s = 1, n = 0 : \tau_1 =$ 

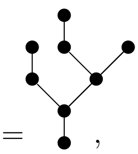
$\tau_2^2 =$  , vanishing tree, discard.

(iii) $\mathbb{T}_3$:

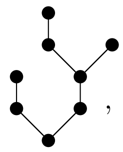
(1) $s = 0, n = 1 : \kappa_1 =$ 

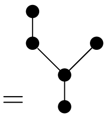
$\tilde{\tau}_1^3 =$  , $\quad \alpha(\tilde{\tau}_1^3) = \frac{1}{3} \cdot 1 \cdot \frac{1}{1!} \cdot \frac{1}{2} = \frac{1}{6};$

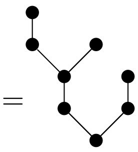
(2) $s = 0, n = 2 : \kappa_1 = \kappa_2 =$ 

$\tilde{\tau}_2^3 =$  , $\quad \alpha(\tilde{\tau}_2^3) = \frac{1}{3} \cdot 1 \cdot \frac{1}{2!} \cdot 1 \cdot 1 = \frac{1}{6};$

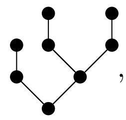
(3) $s = 1, n = 1 : \kappa_1 = \tau_1 =$ 

$\tilde{\tau}_3^3 =$  , $\alpha(\tilde{\tau}_3^3) = \dfrac{1}{3} \cdot (-\dfrac{1}{2}) \cdot \dfrac{1}{1!} \cdot 1 \cdot 1 = -\dfrac{1}{6};$

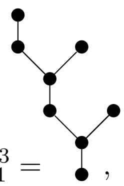
(4) $s = 1, n = 0 : \tau_1 =$ 

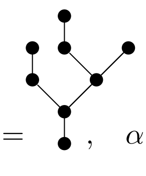
$\tilde{\tau}_4^3 =$  , $\alpha(\tilde{\tau}_4^3) = \dfrac{1}{3} \cdot (-\dfrac{1}{2}) \cdot \dfrac{1}{0!} \cdot 1 \cdot \dfrac{1}{2} = -\dfrac{1}{12};$

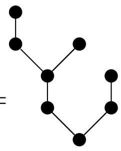
(5) $s = 2, n = 0 : \tau_1 = \tau_2 =$ 

$\tilde{\tau}_5^3 =$  , vanishing tree, discard.

Before we proceed further, let us clean up the set $\mathbb{T}_3$. We clearly see that $\tilde{\tau}_3^3$ is nothing but $\tilde{\tau}_4^3$ with opposite sign. The two trees can be aggregated into $\tilde{\tau}_4^3$ say, with the coefficient replaced by $\alpha(\tilde{\tau}_4^3) - \alpha(\tilde{\tau}_3^3)$. After trivial rotations (corresponding to commutation) we obtain three trees in the set $\mathbb{T}_3$,

$\tau_1^3 =$  , $\alpha(\tau_1^3) = \dfrac{1}{6};$

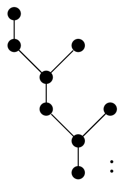
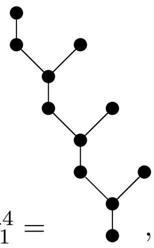
$\tau_2^3 =$  , $\alpha(\tau_2^3) = \dfrac{1}{6};$

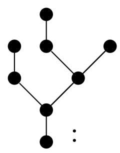
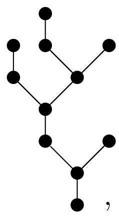
$\tau_3^3 =$  $, \quad \alpha(\tau_3^3) = \dfrac{1}{12}.$

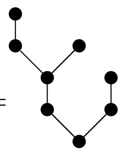
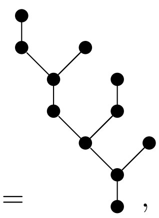
Using the tree formalism, we are now constructing the next "generation" of terms.

(i) $\mathbb{T}_4$:

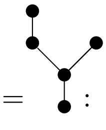
(1) $s = 0, n = 1:$

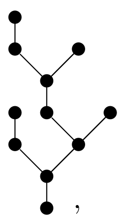
i. $\kappa_1 =$  $: \quad \tilde{\tau}_1^4 =$  $, \quad \alpha(\tilde{\tau}_1^4) = \frac{1}{24};$

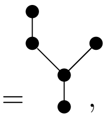
ii. $\kappa_1 =$  $: \quad \tilde{\tau}_2^4 =$  $, \quad \alpha(\tilde{\tau}_2^4) = \frac{1}{24};$

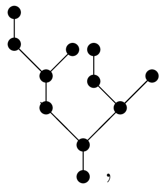
iii. $\kappa_1 =$  $: \quad \tilde{\tau}_3^4 =$  $, \quad \alpha(\tilde{\tau}_3^4) = \frac{1}{48};$

(2) $s = 0, n = 2:$

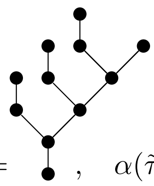
i. $\kappa_1 =$  $\kappa_2 =$  $:$

$\tilde{\tau}_4^4 =$  $, \quad \alpha(\tilde{\tau}_4^4) = \dfrac{1}{16};$

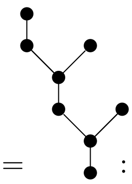
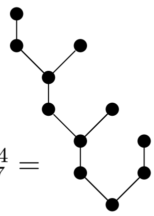
ii. $\kappa_1 =$  , $\kappa_2 =$  :

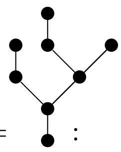
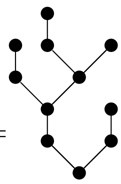
$\tilde{\tau}_5^4 =$  , $\alpha(\tilde{\tau}_5^4) = \dfrac{1}{16}$;

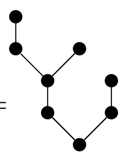
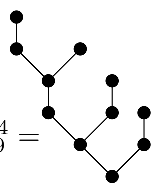
(3) $s = 0, n = 3 : \kappa_1 = \kappa_2 = \kappa_3 =$ 

$\tilde{\tau}_6^4 =$  , $\alpha(\tilde{\tau}_6^4) = \dfrac{1}{24}$;

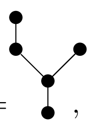
(4) $s = 1, n = 0 :$

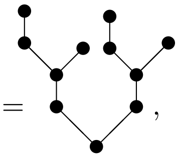
i. $\tau_1 =$  : $\tilde{\tau}_7^4 =$  , $\alpha(\tilde{\tau}_7^4) = -\frac{1}{48}$;

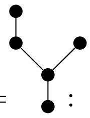
ii. $\tau_1 =$  : $\tilde{\tau}_8^4 =$  , $\alpha(\tilde{\tau}_8^4) = -\frac{1}{48}$;

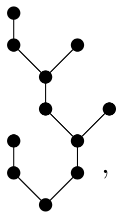
iii. $\tau_1 =$  : $\tilde{\tau}_9^4 =$  , $\alpha(\tilde{\tau}_9^4) = -\frac{1}{96}$;

(5) $s = 1, n = 1 :$

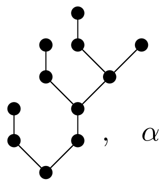
i. $\tau_1 =$  , $\kappa_1 =$  :

$\tilde{\tau}^4_{10} =$  , vanishing tree, discard.

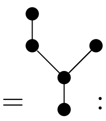
ii. $\tau_1 =$  , $\kappa_1 =$  :

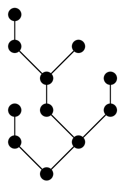
$\tilde{\tau}^4_{11} =$  , $\alpha(\tilde{\tau}^4_{11}) = -\dfrac{1}{16}$;

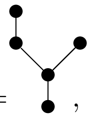
(6) $s = 1, n = 2 : \kappa_1 = \kappa_2 = \tau_1 =$  :

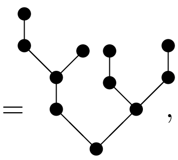
$\tilde{\tau}^4_{12} =$  , $\alpha(\tilde{\tau}^4_{12}) = -\dfrac{1}{16}$;

(7) $s = 2, n = 0 :$

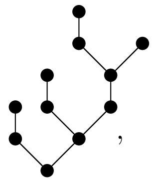
i. $\tau_1 =$  , $\tau_2 =$  :

$\tilde{\tau}^4_{13} =$  , $\alpha(\tilde{\tau}^4_{13}) = \dfrac{1}{96}$;

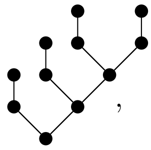
ii. $\tau_1 =$  , $\tau_2 =$  :

$\tilde{\tau}^4_{14} =$  , vanishing tree, discard.

(8) $s = 2, n = 1 : \kappa_1 = \tau_1 = \tau_2 =$  :

$\tilde{\tau}^4_{15} =$  , $\quad \alpha(\tilde{\tau}^4_{15}) = \dfrac{1}{48};$

(9) $s = 3, n = 0 : \tau_1 = \tau_2 = \tau_3 =$  :

$\tilde{\tau}^4_{16} =$  , $\quad \alpha(\tilde{\tau}^4_{16}) = 0;$ vanishing tree, discard.

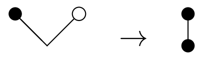
Three trees are vanishing here. Also, with trivial rotations, few trees get cancelled or merge into other trees obeying the anti symmetry, for example, $\tilde{\tau}^4_9$ is $-\tilde{\tau}^4_{13}$ which is also equal to $-(-\tilde{\tau}^4_{15})$, similarly $\tilde{\tau}^4_7$ is nothing but $\tilde{\tau}^4_{11}$ with opposite sign and also $\tilde{\tau}^4_8 = -\tilde{\tau}^4_{12}$ and the trees $\tilde{\tau}^4_3$, $\tilde{\tau}^4_4$ and $\tilde{\tau}^4_5$ satisfy Jacobi identity. Tiding up $\mathbb{T}_4$ and translating every tree in terms of commutators and curly brackets, the Taylor expansion of $\Omega(t)$ becomes

$$
\begin{aligned}
\Omega(t) \;=\; & t\{X_0\} \\
& +\frac{1}{2}t^2\{[\{X_0\},X_0]\} \\
& +t^3(\frac{1}{6}\{[\{[\{X_0\},X_0]\},X_0]\} + \frac{1}{6}\{[\{X_0\},[\{X_0\},X_0]]\} \\
& +\frac{1}{12}[\{[\{X_0\},X_0]\},\{X_0\}]) \\
& +t^4(\frac{1}{24}\{[\{[\{[\{X_0\},X_0]\},X_0]\},X_0]\} \\
& +\frac{1}{24}\{[\{[\{X_0\},[\{X_0\},X_0]]\},X_0]\} + \frac{1}{24}\{[\{X_0\},[\{[\{X_0\},X_0]\},X_0]]\} \\
& +\frac{1}{12}\{[\{[\{X_0\},X_0]\},[\{X_0\},X_0]]\} + \frac{1}{24}\{[\{X_0\},[\{X_0\},[\{X_0\},X_0]]]\} \\
& +\frac{1}{24}[\{[\{[\{X_0\},X_0]\},X_0]\},\{X_0\}] + \frac{1}{24}[\{[\{X_0\},[\{X_0\},X_0]]\},\{X_0\}]) \\
& +\ldots.
\end{aligned}
$$

In explanation of the simplification exercise of the above terms: As we stated that the terms get simplified obeying the anti symmetry and Jacobi identity; By anti symmetry we mean $[A,B] = -[B,A]$, which is an easy exercise to verify for the reader, whereas A, B, C are said to satisfy the Jacobi identity if $[A,[B,C]] + [B,[C,A]] + [C,[A,B]] = 0$. As we translate from the trees,

$$
\begin{aligned}
\tilde{\tau}_3^4 &\rightarrow \{[[\{[\{X_0\}, X_0]\}, \{X_0\}], X_0]\} \rightarrow \{[[\Omega_2, \Omega_1], X_0]\}, \\
\tilde{\tau}_4^4 &\rightarrow \{[\{X_0\}, [\{[\{X_0\}, X_0]\}, X_0]]\} \rightarrow \{[\Omega_1, [\Omega_2, X_0]]\}, \\
\tilde{\tau}_5^4 &\rightarrow \{[\{[\{X_0\}, X_0]\}, [\{X_0\}, X_0]]\} \rightarrow \{[\Omega_2, [\Omega_1, X_0]]\}
\end{aligned}
$$

and hence use the Jacobi identity for simplification. Also, one may observe that by coincidence the tree formation seems identical with [Ise02] by replacing



.

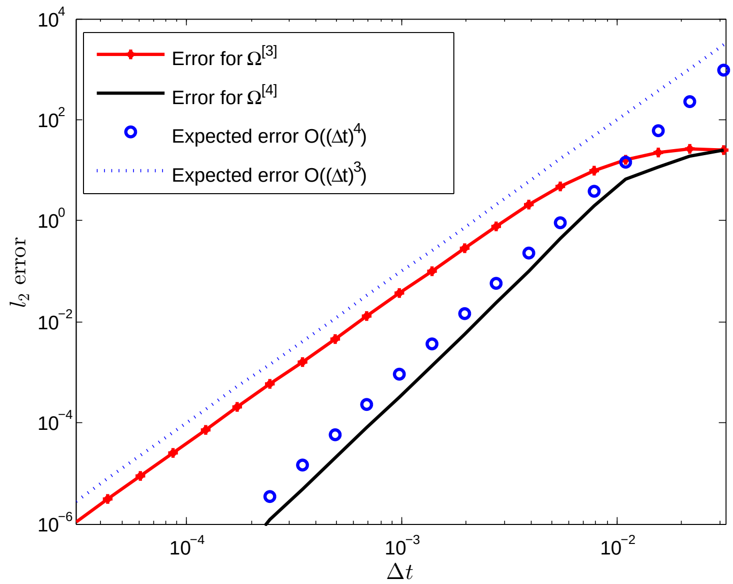


Figure 3.4: Global error on logarithmic scale across an interval [0,1] with different time steps, after truncating the Taylor expansion up to fourth order terms.

It seems quite tempting to introduce such a correspondence between the brackets $\{X\}$ and $[., N]$ for the whole tree exercise. With this replacement we get correct terms up to third order but it gives different values when we reach next generation terms. The reason for this fact is that the difference between $\{X\}$ and $[., N]$ is not only the brackets but the different signs, $+$ and $-$, which appear implicitly in the expression and surely make difference as the complexity increases. Moreover, the number of independent fourth order terms in [Ise02] is 8, whereas the number of independent terms in Ω_4 in BI system is 7. This explains that the fourth order terms obtained from the two approaches are different. Therefore, no such correspondence for the tree structure has been found between the two.

We see that once a tree formulation is obtained, translating it into commutators and curly brackets costs less labour as compared to the complexity of manual computation. After translating the trees into mathematical expressions, we plot the error graph. Truncating the expansion up to fourth order terms, the error graph of the solution as compared to the MATLAB ode45 solver with built-in parameters, is shown in Figure 3.4. The error plot is generated by comparing against the theoretically expected error of $O((\Delta t)^3)$ and $O((\Delta t)^4)$. The experiments were performed on random 25×25 matrices. Clearly, this plot shows that the Magnus method is a fourth order method.

3.4 Concluding remarks

One thus observes that the BI equations have a number of remarkable properties which motivate us to expand the equations. It has been seen in Figure 3.3 that these equations have Lie-Poisson structure. Figure 3.2 shows that the discretisation of the BI equations using Magnus expansion preserves the eigenvalues of the solution matrix. Also, it is observed in Figure 3.4 that the Lie group method using Magnus expansion is a fourth order method, here by fourth order we mean the truncation of $\Omega(t)$ upto the fourth power of t. By employing the shorthand of binary rooted trees for expansion terms, the computation is made affordable. This also lays a foundation to the explicit representation of the solution of the BI system.

Chapter 4

The dynamic generalised double bracket flow

The subject matter of this chapter is generalised double bracket flow, that is, the matrix system of ordinary differential equations of the form

$$X^{'} = [[N, X] + M, X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n) \tag{4.1}$$

where $N \in \mathrm{diag}(n)$ and $M \in \mathfrak{so}(n)$. We are concerned with analysing and discretising (4.1). Clearly, we see that it is of the form

$$X^{'} = [B(X), X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n),$$

where $B(X) : \mathrm{Sym}(n) \to \mathfrak{so}(n)$ is given by $B(X) = [N, X] + M$. This is a form of an isospectral flow.

Further we observe that when $M = 0$, it reduces to the double bracket flow, which is explained in detail in section 4.1. Moreover, when $N = 0$, this reduces to an integrable system. Now the question arises as to how it behaves when both N and M are non-zero. In this chapter we analyse the dynamics of (4.1) for different values of N and M. We present the phase portraits for various non-zero values of N and M. Before proceeding further let us have a brief look at the double bracket flow in the following section.

4.1 Double bracket flow

As discussed above, when $M = 0$, (4.1) reduces to the double bracket flow (dbf)

$$X^{'} = [[N, X], X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n).$$

Dbf was introduced by Brockett in [Bro91], and Chu and Driessel in [CD90]. The double bracket equations possess a number of features. First of all, it is a particular example of an isospectral flow [Ise02].

$$X^{'} = [A(t, X), X], \quad X(0) = X_0 \in \mathrm{Sym}(n),$$

where $A \in \mathfrak{so}(n)$, the Lie-algebra of $n \times n$ real skew-symmetric matrices. Then there exists a matrix function $Q(t) \in \mathrm{SO}(n)$ such that

$$X(t) = Q(t)X_0Q^T(t)$$

and $X(t)$ has the same eigenvalues as X_0 for all $t \geq 0$.

The second feature of dbf is that it is a gradient system, with a global Lyapunov function, therefore it is assured of convergence to a fixed point of the flow as $t \to 0$ [Bro91]. Given the potential function

$$\psi(X) = \frac{1}{2}\|X - N\|_F^2$$

where $\|.\|_F$ is the Frobenius norm and X ranges across all symmetric matrices orthogonally similar to X_0, then it is shown in [CD90] that $\nabla\psi(X) = -[[N, X], X]$ and thus dbf is precisely gradient system $X^{'} = -\nabla\psi(X)$. Hence, the double-bracket flow minimizes ψ. $X_\infty = \lim_{t\to\infty} X(t)$ is a local minimizer of ψ in [Bro91].

We can use the double-bracket flow to diagonalize real symmetric matrices, and to find their eigenvalues. It has been shown by Brockett that when N is a real diagonal matrix and both X_0 and N have distinct eigenvalues, then $X(t)$ tends exponentially to a diagonal matrix as $t \to +\infty$ and the eigenvalues are sorted accordingly to the diagonal entries of N.

There are other applications including sorting lists and solving certain linear programming problems [Blo90].

In the double bracket flow, different choices of N correspond to special realization processes. Therefore, the appealing part of the dbf is also the flexibility caused by its dependence on the matrix N. If $N = diag(1, 2, \ldots, n)$ and Y is tridiagonal, then dbf gives the Toda flow on tridiagonal matrices [Blo90].

Double-bracket equation has been well understood from a theoretical point of view. When we talk about efficiently computing the solutions of dbf, we know there are several numerical methods, including the family of Lie-group methods [IMKNZ99] and [Zan97]. The idea is to write the solution in the form $X(t) = Q(t)X_0Q^T(t)$ and, instead of computing X directly, in each step an orthogonal matrix Q_{k+1} is evaluated, so that $X_{k+1} = Q_{k+1}X_kQ_{k+1}^T$. The matrix Q_{k+1} is chosen as the solution of the initial value problem

$$Q_{k+1}^{'} = A(t, Q_{k+1}X_kQ_{k+1}^T)Q_{k+1}, Q_{k+1}(kh) = I$$

which is a particular example of a Lie-group flow. In [Ise02], the dbf is represented in the form $X(t) = exp(\Omega(t))X_0 exp(-\Omega(t))$ and the Taylor expansion of Ω is constructed explicitly identifying individual expansion terms with certain rooted trees with bicolour leaves.

In the following section, we proceed to the analysis of generalised double bracket flow (4.1), and we compute their solutions in the next chapter.

4.2 Analysing the generalised double bracket flow

In this section we analyse the given system (4.1). As mentioned earlier, when $N = 0$, (4.1) reduces to an integrable system and when $M = 0$ it reduces to the double bracket flow. We will see how the dynamics change with different values of M and N. We also do the stability analysis taking examples of 2×2 and 3×3 matrices.

4.2.1 Analysing taking 2×2 matrices

Taking

$$X_0 = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, N = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ and } M = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

we get

$$X = \begin{bmatrix} \frac{1}{2}(a+c) - b\sin 2t + \frac{1}{2}(a-c)\cos 2t & b\sin 2t + \frac{1}{2}(a-c)\cos 2t \\ b\sin 2t + \frac{1}{2}(a-c)\cos 2t & \frac{1}{2}(a+c) + b\sin 2t - \frac{1}{2}(a-c)\cos 2t \end{bmatrix}$$

which is an oscillatory matrix.

Taking

$$X = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \quad N = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \text{ and } M = \begin{bmatrix} 0 & \alpha \\ -\alpha & 0 \end{bmatrix}$$

we get

$$X' = \begin{bmatrix} 2b((d_1-d_2)b+\alpha) & (c-a)((d_1-d_2)b+\alpha) \\ (c-a)((d_1-d_2)b+\alpha) & -2b((d_1-d_2)b+\alpha) \end{bmatrix}$$

that is

$$\begin{aligned} a' &= 2b((d_1-d_2)b+\alpha) \\ b' &= (c-a)((d_1-d_2)b+\alpha) \\ c' &= -2b((d_1-d_2)b+\alpha) \end{aligned} \tag{4.2}$$

Solving the equations (4.2)

$$a^{'} + c^{'} = 0$$

or $a + c =$ constant

In particular, we assume $a + c = 0, \quad a = -c$ then

$$\begin{aligned} a^{'} &= 2b((d_1 - d_2)b + \alpha) \\ b^{'} &= -2a((d_1 - d_2)b + \alpha) \end{aligned} \tag{4.3}$$

$$\Rightarrow \frac{a^{'}}{b^{'}} = \frac{b}{-a}$$

or

$$-aa^{'} = bb^{'}$$

or $a^2 + b^2 = \gamma^2$ where $\gamma^2 = \sqrt{a_0^2 + b_0^2}$. Therefore

$$X = \begin{bmatrix} a & \sqrt{\gamma^2 - a^2} \\ \sqrt{\gamma^2 - a^2} & -a \end{bmatrix}$$

Taking $a = \gamma \cos\theta$, we have

$$X = \sqrt{a_0^2 + b_0^2} \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}$$

We can see that this is a circle. To find the stability of the equations, we will use the Jacobian matrix

Now, fixed points of these equations are

(i) $b = 0, a = c$,

(ii) $b = \frac{-\alpha}{d_1 - d_2}$.

We know that equilibrium is stable if all eigenvalues have negative real part, it is unstable if at least one eigenvalue has positive real part. Jacobian is given as

$$J = \begin{bmatrix} 0 & 2((d_1 - d_2)b + \alpha) + 2b(d_1 - d_2) & 0 \\ -((d_1 - d_2)b + \alpha) & (c - a)(d_1 - d_2) & ((d_1 - d_2)b + \alpha) \\ 0 & -2((d_1 - d_2)b + \alpha) - 2b(d_1 - d_2) & 0 \end{bmatrix}$$

Here, the determinant of Jacobian matrix is always zero. So, at least one of the eigenvalues is zero. Then the origin is not an isolated fixed point.

Finding the eigenvalues of the Jacobian matrix at:

(i) $b = 0, a = c$

$$J = \begin{bmatrix} 0 & 2\alpha & 0 \\ -\alpha & 0 & \alpha \\ 0 & -2\alpha & 0 \end{bmatrix}$$

Characteristic equation is given by $JI - \lambda I = 0$

$$\begin{vmatrix} -\lambda & 2\alpha & 0 \\ -\alpha & -\lambda & \alpha \\ 0 & -2\alpha & -\lambda \end{vmatrix} = 0$$

$$\lambda = 0, \quad \pm 2\alpha\iota, \text{ where } \iota = \sqrt{-1}.$$

We see that system is unstable at the origin. It is stable for $\lambda < 0$. The nature of this equilibrium point says non hyperbolic fixed point is center.

(ii) $b = -\frac{\alpha}{d_1 - d_2}$

$$J = \begin{bmatrix} 0 & -2\alpha & 0 \\ 0 & (c-a)(d_1 - d_2) & 0 \\ 0 & 2\alpha & 0 \end{bmatrix}$$

Characteristic equation is given by $JI - \lambda I = 0$

$$\begin{vmatrix} -\lambda & -2\alpha & 0 \\ 0 & (c-a)(d_1 - d_2) - \lambda & 0 \\ 0 & 2\alpha & -\lambda \end{vmatrix} = 0$$

$$\lambda = 0, \quad \lambda = (c-a)(d_1 - d_2)$$

For stability we should have $\lambda = (c-a)(d_1 - d_2) < 0$, That means one of $c - a$ and $d_1 - d_2$ should be negative for stability. This is a non hyperbolic fixed point.

The dynamics of the model system (4.2) is showing stable limit cycle. The solution is presented as time series in Figure 4.1 and 2D and 3D phase plots showing the portraits with respect to $(a,b),(a,c),(b,c)$ and (a,b,c) respectively in the Figures 4.2(i), 4.2(ii), 4.2(iii) and 4.2(iv). These plots are generated taking the initial condition

$$X_0 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

.

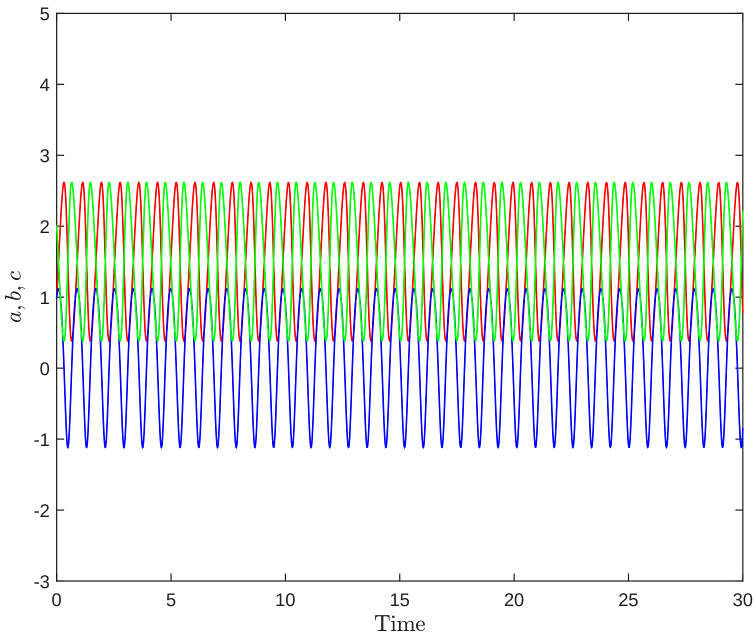


Figure 4.1: Showing time series of the model system (4.2) with $d_1 = 4, d_2 = 5, \alpha = 4$.

4.2.2 An explicit solution taking 2×2 matrices

We also attempt to find an explicit solution for (4.1) taking 2×2 matrices. Having again a look at the above system (4.2), we denote,

$$\delta = d_1 - d_2 \neq 0$$

therefore rewriting (4.2),

$$\begin{aligned} a' &= 2b(\delta b + \alpha) \\ b' &= (c - a)(\delta b + \alpha) \end{aligned}$$

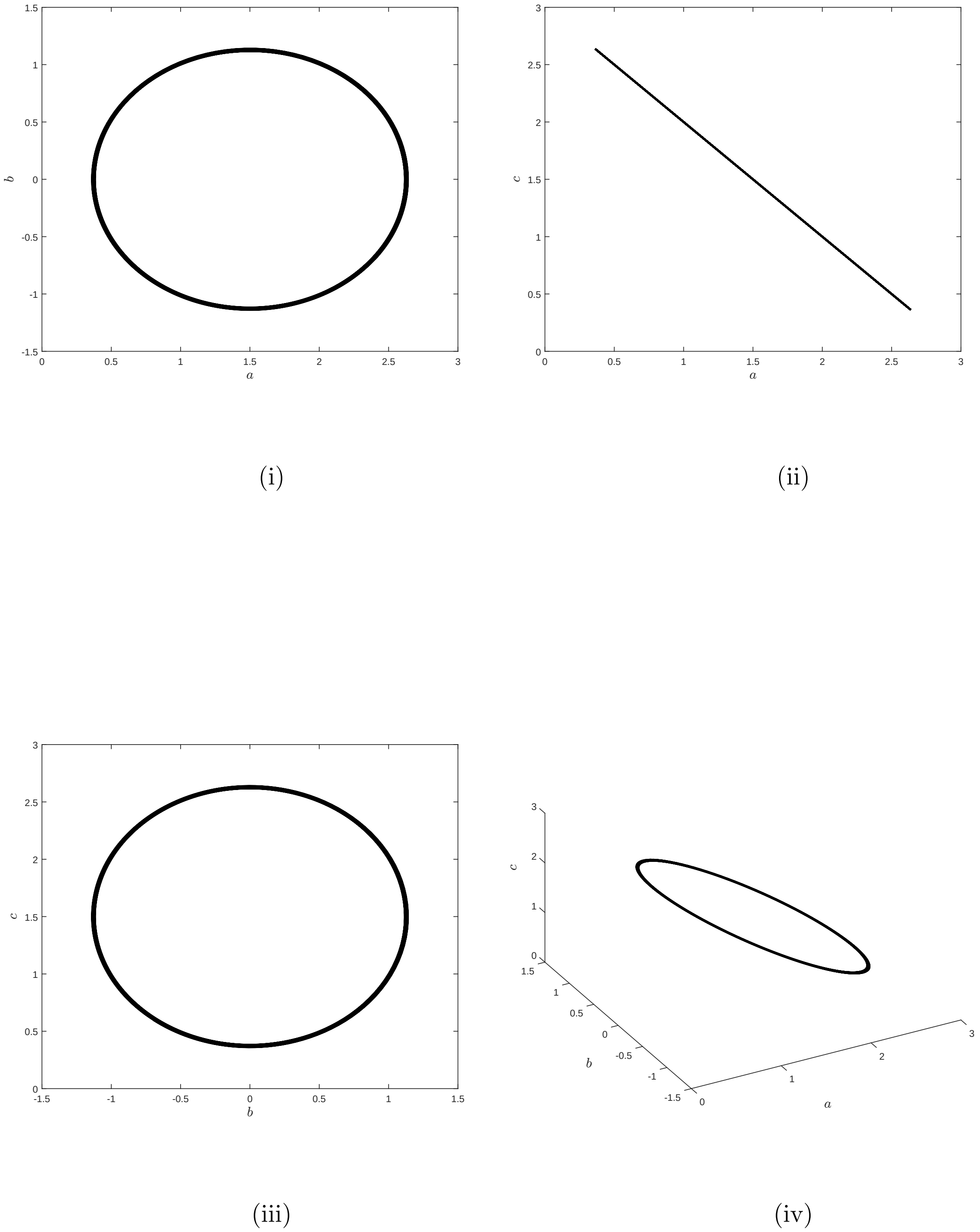


Figure 4.2: Phase plots $(a, b), (a, c), (b, c)$ and (a, b, c), respectively, of the model system (4.2) with $d_1 = 4, d_2 = 5, \alpha = 4$.

$$c' = -2b(\delta b + \alpha)$$

$\therefore a' + c' = 0$ and $a + c = const.$

Let $p(t) = a(t) - c(t)$, $q(t) = \delta b(t) + \alpha$; [or $b(t) = \frac{q(t)-\alpha}{\delta}$]

$$\begin{aligned} \therefore p' &= a' - c' \\ &= 4b(\delta b + \alpha) \\ &= 4(\frac{q(t) - \alpha}{\delta}).q \\ &= \frac{4}{\delta}q(q - \alpha) \end{aligned}$$

Similarly,

$$\begin{aligned} q' &= \delta b' \\ &= \delta(c - a)(\delta b + \alpha) \\ &= \delta pq \end{aligned}$$

The general solution with MAPLE and trivial simplifications is

$$\begin{aligned} p(t) &= -\frac{c_1[(64\alpha^2 - 16c_1^2)e^{2c_2+2+c_1} - 1]}{\delta[(8\alpha e^{tc_1+c_2} + 1)^2 - 16c_1^2 e^{2(tc_1+c_2)}]} \\ q(t) &= \frac{4c_1^2 e^{tc_1+c_2}}{(8\alpha e^{tc_1+c_2} + 1)^2 - 16c_1^2 e^{2(tc_1+c_2)}} \end{aligned}$$

(i) When $c_1 > 0$
$\underset{t\to\infty}{lim}\ q(t) = 0,\ \underset{t\to\infty}{lim}\ p(t) = -\frac{c_1}{\delta}$
($c_1 \neq 4\alpha$)

(ii) When $c_1 < 0$
$\underset{t\to\infty}{lim}\ q(t) = 0,\ \underset{t\to\infty}{lim}\ p(t) = \frac{c_1}{\delta}$

For $e^{c_2} \to c_2$, we get the solution as

$$\begin{aligned} p(t) &= -\frac{c_1[(64\alpha^2 - 16c_1^2)c_2 e^{2tc_1} - 1]}{\delta[(8\alpha c_2 e^{tc_1} + 1)^2 - 16c_1^2 c_2^2 e^{2tc_1}]} \\ q(t) &= \frac{4c_1^2 c_2 e^{tc_1}}{(8\alpha c_2 e^{tc_1} + 1)^2 - 16c_1^2 c_2^2 e^{2tc_1}} \end{aligned}$$

At t = 0, the solution is

$$p(0) = -\frac{c_1[64\alpha^2 c_2^2 - 16c_1^2 c_2^2 - 1]}{\delta[(8\alpha c_2 + 1)^2 - 16c_1^2 c_2^2]}$$
$$q(0) = \frac{4c_1^2 c_2}{(8\alpha c_2 + 1)^2 - 16c_1^2 c_2^2}$$

$\therefore\ c_1 =$ zeroes of $Z^2 = \lambda^2 p_0^2 + 8\alpha q_0 - 4q_0^2$

$$\therefore\ c_1 = \pm\sqrt{\lambda^2 p_0^2 + 8\alpha q_0 - 4q_0^2}$$
$$c_2 = \frac{\frac{q_0}{2}}{\pm\lambda p_0\sqrt{\lambda^2 p_0^2 + 8\alpha q_0 - 4q_0^2} + \lambda^2 p_0^2 + 4\alpha q_0 - 4q_0^2}$$

Hence, we have three cases, firstly, when we have $\lambda^2 p_0^2 + 8\alpha q_0 - 4q_0^2 \geq 0$ this implies $p(t), q(t) \to 0$. In the second case, if $\lambda^2 p_0^2 + 8\alpha q_0 - 4q_0^2 < 0$, the system reduces to a limit cycle. And thirdly, in case $\lambda^2 p_0^2 + 8\alpha q_0 = 4q_0^2$, this results in a Hopf bifurcation and vice versa. In Hopf bifurcation the real parts of a pair of complex conjugate eigenvalues of the least-stable equilibrium point increase through zero as a control parameter is varied through a critical value. A time-periodic solution may arise after bifurcation.

4.3 Some interesting phase portraits

In the last section of this chapter, we present some phase portraits showing the remarkable behaviour of generalised double bracket flows. We have computed numerically the solution of the system for random 3×3 matrices using Lie group method using Magnus expansion. In the Figures 4.3–4.6, the phase portraits $(X_{1,2}, X_{k,l})$ are displayed for $(k, l) = (1,1), (1,3), (2,2), (2,3), (3,3)$, with random initial condition. Consider the matrices

$$N = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \text{ and } M = \begin{bmatrix} 0 & h & g \\ -h & 0 & l \\ -g & -l & 0 \end{bmatrix}$$

We discuss the following cases-

We fix the matrices choosing $d_1 = 1, d_2 = 2, d_3 = 3$ and $h = 2, g = 3, l = 1$. Setting a parameter p where $p \in (0, 1)$, we generate the phase portraits for $p \cdot N$ and $(1 - p) \cdot M$ for different values of p. Plots are generated for $p = 0.9,\ 0.7,\ 0.1,$ and 0.0001, respectively in the Figures 4.3, 4.4, 4.5 and 4.6. In figure 4.6, each string is again a bunch of strings, it is shown in the last plot in Figure 4.6 by focusing on the enlarged image in one of the plots. Similar behaviour (as in 4.6) is seen for other values of the form $p = 10^{-n}$ where $n = \{1, 2, 3, \cdots\}$.

We see Hopf bifurcation occurs in a periodic orbit of an autonomous flow. In this case the invariant curve corresponds to an invariant torus for the flow and attracting periodic orbits on the circle correspond to mode-locked periodic motion on the torus, whilst dense orbits correspond to quasi-periodic motion.

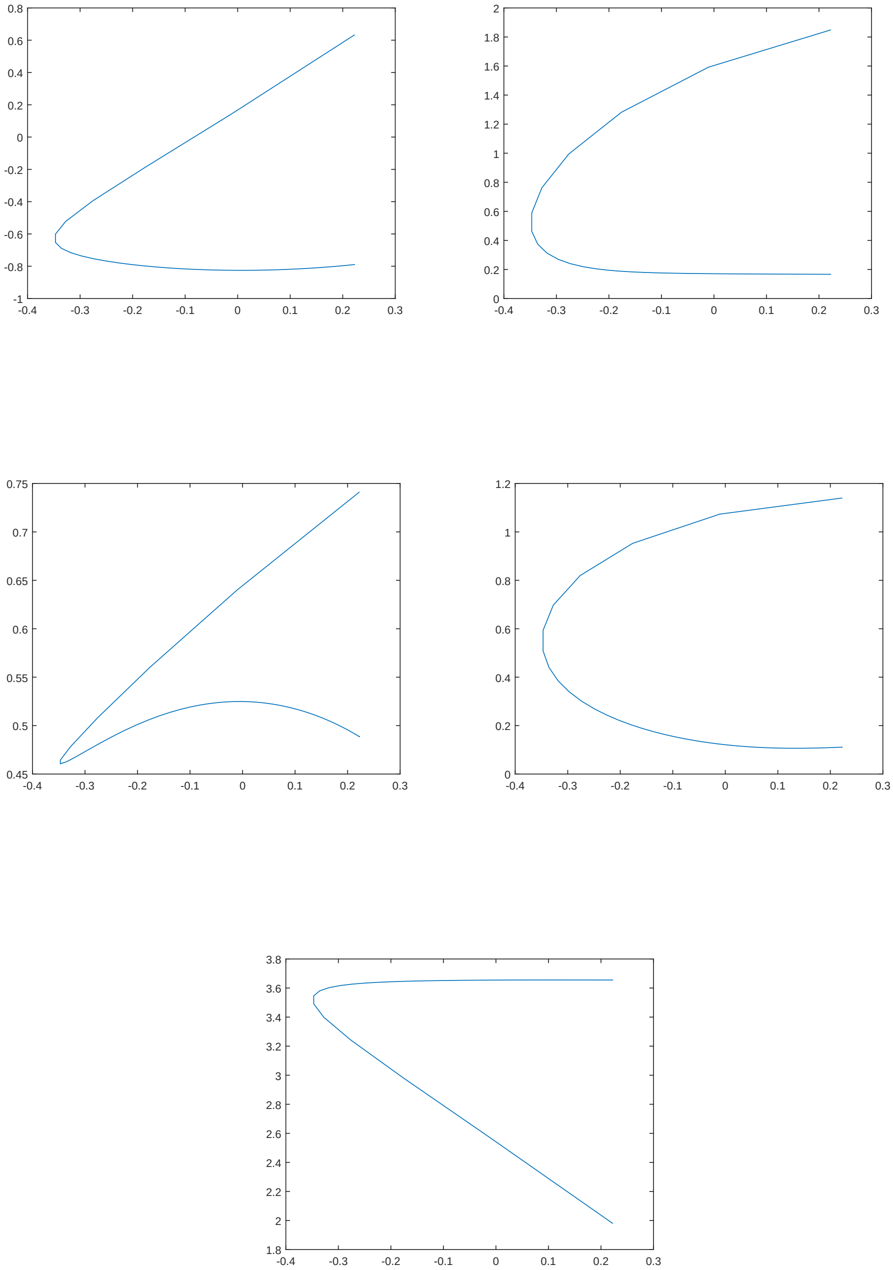


Figure 4.3: The phase portraits $(X_{1,2}, X_{k,l})$ for $(k, l) = (1, 1), (1, 3), (2, 2), (2, 3), (3, 3)$, respectively, with a random initial condition. Here by $X_{k,l}$ we mean the kl^{th} element of the matrix X. These plots are generated choosing $d_1 = 1, d_2 = 2, d_3 = 3$ and $h = 2, g = 3, l = 1$ and $p = 0.9$.

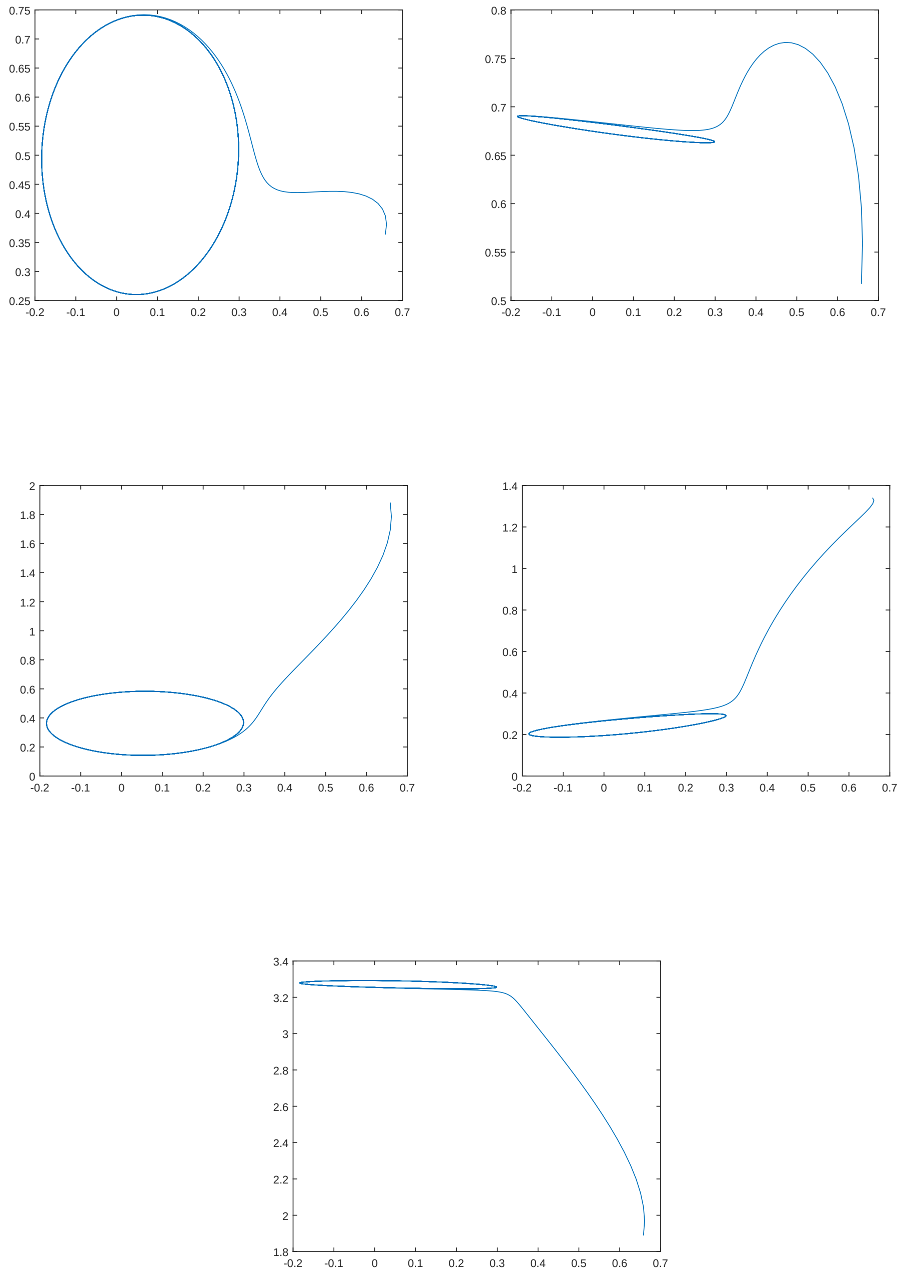


Figure 4.4: The phase portraits $(X_{1,2}, X_{k,l})$ for $(k,l) = (1,1), (1,3), (2,2), (2,3), (3,3)$, respectively, with a random initial condition. Here by $X_{k,l}$ we mean the kl^{th} element of the matrix X. These plots are generated choosing $d_1 = 1, d_2 = 2, d_3 = 3$ and $h = 2, g = 3, l = 1$ and $p = 0.7$.

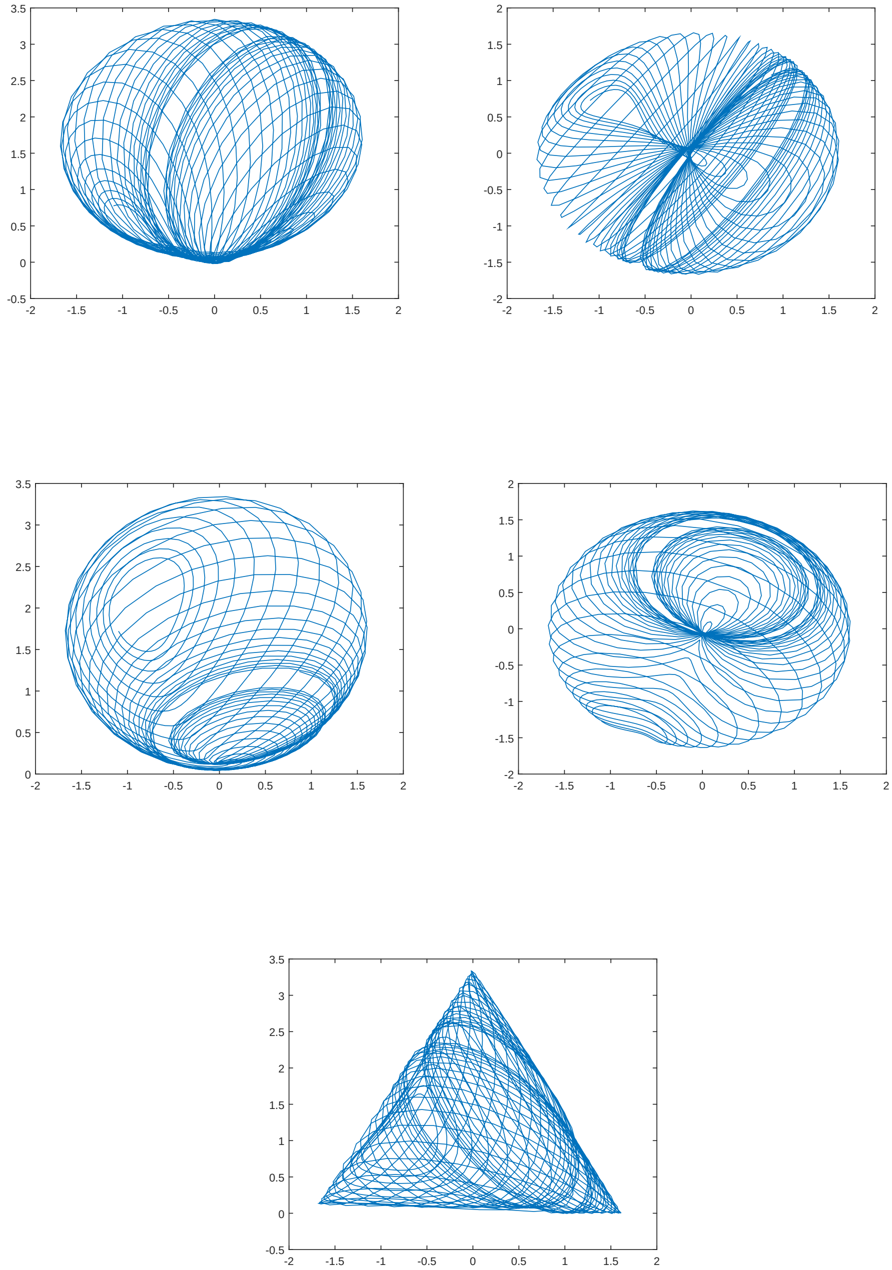


Figure 4.5: The phase portraits $(X_{1,2}, X_{k,l})$ for $(k,l) = (1,1), (1,3), (2,2), (2,3), (3,3)$, respectively, with a random initial condition. Here by $X_{k,l}$ we mean the kl^{th} element of the matrix X. These plots are generated choosing $d_1 = 1, d_2 = 2, d_3 = 3$ and $h = 2, g = 3, l = 1$ and $p = 0.1$.

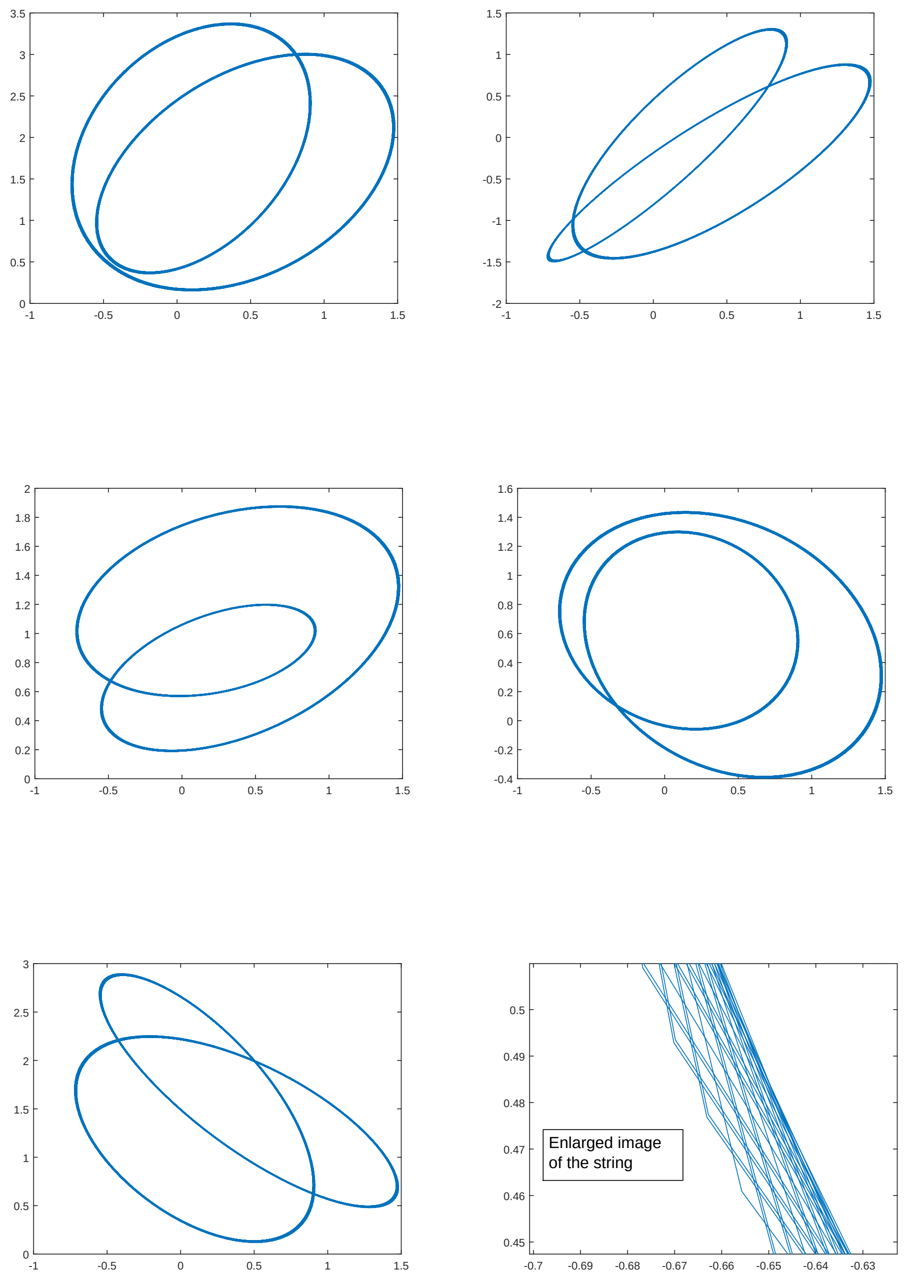


Figure 4.6: The phase portraits $(X_{1,2}, X_{k,l})$ for $(k,l) = (1,1), (1,3), (2,2), (2,3), (3,3)$, respectively, with a random initial condition. Here by $X_{k,l}$ we mean the kl^{th} element of the matrix X. These plots are generated choosing $d_1 = 1, d_2 = 2, d_3 = 3$ and $h = 2, g = 3, l = 1$ and $p = 0.0001$.

4.4 Concluding remarks

Thus, we observe the phase portraits of the generalised double bracket flow for various values of M and N. We observe that, for $M_{ij} = 0, \quad X(t) \to \hat{X}$ (this is the case of dbf), and for $M_{ij} \gg 1, \quad X(t) \to \pi$ as the system has Hopf bifurcation and gives birth to the limit cycles. Limit cycles have been used to model the behaviour of many oscillatory systems. The Hopf bifurcation underlies many spontaneous oscillations such as airfoil flutter and other wind-induced oscillations (e.g., those that caused the Tacoma-Narrows bridge collapse) in structural engineering systems, vortex shedding in fluid flow around a solid body at sufficiently high stream velocity, LCR oscillations in electrical circuits, relaxation oscillations (e.g., as described by the Van der Pol oscillator), the periodic firing of neurons in nervous systems (e.g., in the FitzHugh-Nagumo equation modelling these phenomena), oscillations in autocatalytic chemical reactions (e.g., the Belousov-Zhabotinsky reaction) as described by the Brusselator and similar models, oscillations in fish populations (as described by predator-prey models), periodic fluctuations in the number of individuals suffering from an infectious disease (as described by epidemic models), etc. For references see, [MM12] [VDM85] [Sac64].

Having seen the interesting applications of Hopf bifurcation, we could study the analysis of this system more in detail in the sequel. In the next chapter we move towards discretising the flow using Magnus series.

Chapter 5

Discretisation of generalised double bracket flow

This chapter extends the method of Magnus expansion to Lie-algebraic equations in the generalised double bracket flow (gdbf) (4.1). The idea is to write the solution of (4.1) in the form $X(t) = e^{\Omega(t)}X_0e^{-\Omega(t)}$, where instead of computing X at the first place, we obtain the Taylor expansion of Ω. Our goal is to determine the rules for finding the terms of Ω to an arbitrary accuracy.

Following the same solution structure as we did for BI equations, first we convert the isospectral flow to a Lie-group flow and then translate it into a Lie-algebraic equation. This method preserves the isospectrality and gives the desired structure of the solution with large time steps. We solve the given system of differential equations using the Magnus expansion to obtain the Taylor expansion of Ω. The terms are represented by binary rooted trees with tri-colour leaves and an algorithm is formed to construct the next tree by recursion and to calculate the coefficient of each tree. The explicit representation of the solution of the given equation is derived and $B(X)$ is represented in a finite "alphabet". The representation as binary trees is very important because, as the number of terms in each iteration grows exponentially, the complexity of manual computation becomes prohibitive. By indexing the terms in the expansion with a subset of binary trees, it is convenient to derive explicit recurrence relations.

We organise this chapter as follows. In section 5.1 we expand the solution of (4.1) using Magnus series and represent it in the form $X(t) = e^{\Omega(t)}X_0e^{-\Omega(t)}$, which ensures that the solution structure remains isospectral. Taylor expansion of Ω has been formed algorithmically from X_0, N, M and linear combinations of their commutators. The graph for solution error and the error in eigenvalues are displayed. We determine the precise rules underpinning this process and obtain explicit numerical procedure for the approximation of Ω to arbitrary accuracy. The procedure to construct Taylor expansion of Ω is similar using the Magnus expansion and it employes the terminology of binary rooted trees. This is much more complicated and delicate because of the three "alphabet", X_0, N, M, and their commutators. In the next section 5.2 the terms are represented by binary rooted trees with tri-colour leaves and we acquire a rule to construct the next tree by recursion and to calculate the coefficient of each tree. Although this matrix system appears more complex and leads to the tri-colour leaves; it has been possible to formulate the explicit recursive rule. In the case of this matrix system, we have two types of trees and an interplay between them, which makes it more convoluted and difficult. A step by step algorithm is developed and recurrence formula is defined that enables us to compute the trees and coefficients explicitly. In section 5.2.1 we assemble all the information into a well defined numerical algorithm. Using this algorithm next generation trees are constructed and their coefficient are calculated in section 5.2.2. Translating the trees into X_0, N, M and their commutators gives us the required expansion of Ω.

5.1 Expansion of the solution

We know (see [IMKNZ99]) that no classical numerical methods can respect the Lie–group structure. Exceptionally, in case of orthogonal flows, symplectic Runge–Kutta methods preserve orthogonality [LD94]. However, because of being implicit, it is expensive to use those methods. We use the method which respects the structure. Rewriting (4.1) to discretise using Magnus expansion

$$X^{'} = [[N,X]+M,X], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n)$$

$X^{'} = [B(X),X], \quad B(X) = [N,X]+M$ where $B(X) : \mathrm{Sym}(n) \to \mathfrak{so}(n)$. It is standard to verify that

$$X(t) = Q(t)X_0Q^T(t), \quad t \geq 0,$$

where $Q(t) \in \mathrm{SO}(n)$ is the solution of

$$Q'(t) = B(Q(t)X_0Q^T(t))Q(t), \quad Q(0) = I. \tag{5.1}$$

In a similar way as we did for Bloch–Iserles equations in chapter 2, our idea is to represent the solution of the linear equation (5.1) in the form

$$Q(t) = e^{\Omega(t)},$$

where

$$\Omega' = \sum_{0}^{\infty} \frac{\mathrm{B}_r}{r!}\mathrm{ad}_{\Omega}^{r}([N, e^{\Omega}X_0e^{-\Omega}]+M), \quad \Omega(0) = 0. \tag{5.2}$$

Here $\mathrm{B}_m,\ m \in \mathbb{Z}$ are Bernoulli numbers and ad_{Ω}^{r} is an iterated commutator defined by

$$\mathrm{ad}_{\Omega}^{0}A = A, \quad \mathrm{ad}_{\Omega}^{1}A = [\Omega, A], \quad \mathrm{ad}_{\Omega}^{2}A = [\Omega, [\Omega, A]], ..., \quad \mathrm{ad}_{\Omega}^{m}A = [\Omega, \mathrm{ad}_{\Omega}^{m-1}A],$$

where $[\Omega, A] = \Omega A - A\Omega$.

Now, taking $\Omega(t) = \sum_{m=0}^{\infty} \Omega_m t^m$ gives

$$\Omega'(t) = \sum_{m=0}^{\infty} (m+1)\Omega_{m+1}t^m \tag{5.3}$$

and this implies

$$\sum_{m=0}^{\infty} (m+1)\Omega_{m+1}t^m = \sum_{r=0}^{\infty} \frac{\mathrm{B}_r}{r!} \mathrm{ad}_{\Omega(t)}^{r}([N, e^{\Omega}X_0e^{-\Omega}] + M). \tag{5.4}$$

Comparing coefficients of $t^0, t^1, t^2...$ we get the values of $\Omega_1, \Omega_2, \Omega_3...$ as follows

$$\begin{aligned}
\Omega_1 &= [N, X_0] + M, \\
\Omega_2 &= \frac{1}{2}[N, [\Omega_1, X_0]], \\
\Omega_3 &= \frac{1}{3}[N, [\Omega_2, X_0]] + \frac{1}{6}[N, [\Omega_1, [\Omega_1, X_0]]] \\
&- \frac{1}{6}[\Omega_1, \Omega_2].
\end{aligned}$$

It is time to analyse the error graphs of the solution that is obtained using Lie group method. Before simplifying the above expansion, we present the preliminary error graph and error graph for eigenvalues of this method, as compared to the MATLAB ode45 solver with built-in parameters, in Figure 5.1 and Figure 5.2, respectively.

In Figure 5.1 we display error in the solution of gdbf in the interval [0,1] for a range of different step sizes Δt. The error plot is generated by truncating the expansion up to order three and is compared against the theoretically expected error of $O((\Delta t)^3)$. The experiments were performed on random matrices. In Figure 5.2 we calculate absolute error of eigenvalues of the two methods on a logarithmic scale. It is clearly seen that our method preserves the correct eigenvalues to machine accuracy. We calculated the error in eigenvalues with small and large time steps. Despite a large time-step in the Magnus method, the error in eigenvalues stays very close to machine precision while the solution obtained using ode45 quickly strays away in terms of eigenvalues as time increases. Which is the expected behaviour of Magnus method from the principles underlying our approach.

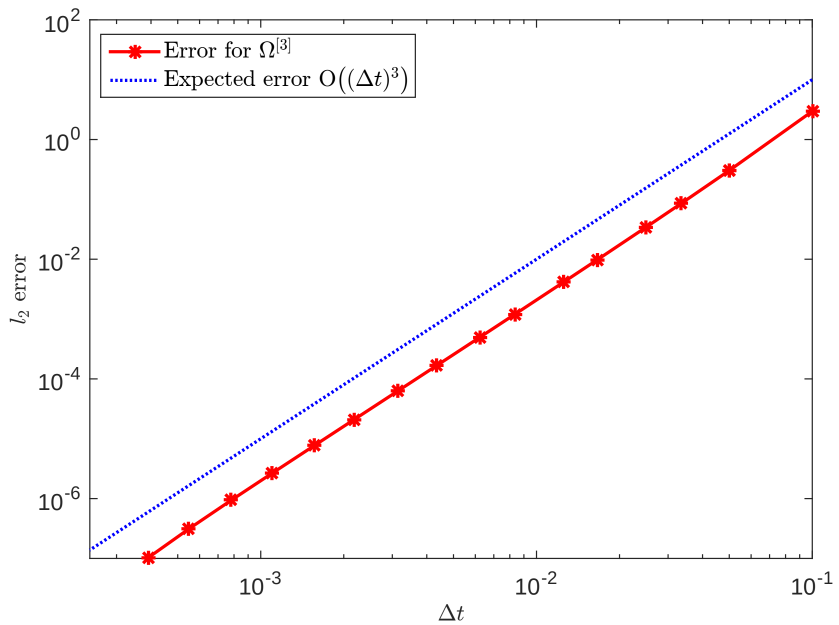


Figure 5.1: Global error on logarithmic scale across an interval [0,1] with different time steps, after truncating the Taylor expansion up to third order terms.

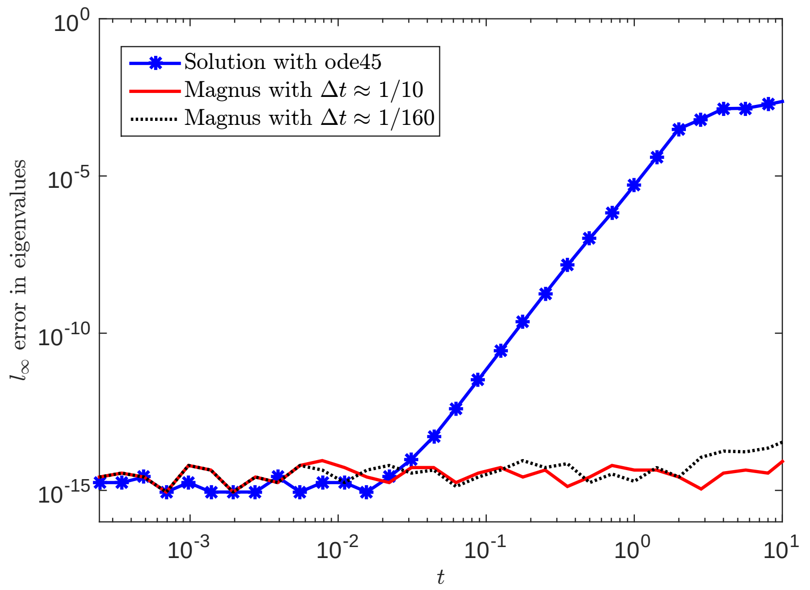


Figure 5.2: Error plot showing absolute error of eigenvalues of the two methods on a logarithmic scale.

5.2 Representation by binary trees

Let us again look at the expansion Ω, we note that each term is written in just three 'letters' X_0, N and M, hence belongs to a free structure generated by them. This seems relatively complex than the expansion of Bloch–Iserles equations in chapter 2, because there is an extra 'letter' M and it appears implicitly in every expression and making it more complex. To understand the expansion terms and the tree structure in a better way we tend to separate the commutators in every term when it comes with $+M$ e.g. $[., X+M] = [., X] + [., M]$, that for sure is going to increase the number of trees for every term. We attempt to find the expansion of the solution in Taylor series of the form

$$\Omega(t) = \sum_{r=1}^{\infty} t^r \sum_{\tau \in \mathbb{T}_r} \alpha(\tau) H_\tau \tag{5.5}$$

where $\mathbb{T}_r$ is the set of all binary trees of power r, H_τ is an expression constructed from X_0, N and M according to rules implicit in the structure of the tree τ which will be explained next, and α is a scalar constant. We use binary rooted trees as a shorthand for expansion terms, an approach introduced by [IN99], also used for BI equations in chapter 3, that leads to a groundwork illuminating the structure of individual terms and their relationship. We choose to assign to the three 'letters', X_0, N and M, the leaves in three different colours namely, black, white, and black-n-white respectively,

$$\bullet \rightsquigarrow \; X_0.$$

and

$$\circ \rightsquigarrow \; N, \text{and} \quad \odot \rightsquigarrow M.$$

We define a function $\tau \to H_\tau$ from $\mathbb{T} = \bigcup_{r=1}^{\infty} \mathbb{T}_r$, a subset of binary rooted trees into $n \times n$ matrix functions by letting $H_\bullet = X_0$ and, by induction,

$$\begin{array}{c} \tau_1 \quad \tau_2 \\ \diagdown\!\diagup \end{array} \rightsquigarrow \; [H_{\tau_1}, H_{\tau_2}],$$

where H_{τ_1} and H_{τ_2} are already constructed expansion terms.

We delve into the general convention for the correspondence between trees and expansion terms. Let us have again a look at the equation

$$\sum_{m=0}^{\infty} (m+1)\Omega_{m+1} t^m = \sum_{r=0}^{\infty} \frac{\mathrm{B}_r}{r!} \mathrm{ad}_{\Omega(t)}^r \Big([N, e^{\Omega(t)} X_0 e^{-\Omega(t)}] + M \Big).$$

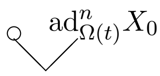
We know that

$$e^{\Omega}X_0e^{-\Omega} = \mathrm{Ad}_{\Omega}X_0 = e^{\mathrm{ad}_{\Omega}}X_0 = \sum_{n=0}^{\infty}\frac{1}{n!}\mathrm{ad}^n_{\Omega}X_0$$

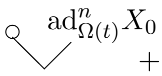
Thus

$$\sum_{m=0}^{\infty}(m+1)\Omega_{m+1}t^m = \sum_{r=0}^{\infty}\frac{\mathrm{B}_r}{r!}\mathrm{ad}^r_{\Omega(t)}\Big([N,\sum_{n=0}^{\infty}\frac{1}{n!}\mathrm{ad}^n_{\Omega(t)}X_0]+M\Big)$$

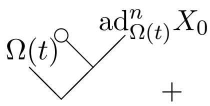
$$= \sum_{r=0}^{\infty}\frac{\mathrm{B}_r}{r!}\mathrm{ad}^r_{\Omega(t)}[N,\sum_{n=0}^{\infty}\frac{1}{n!}\mathrm{ad}^n_{\Omega(t)}X_0] + \sum_{r=0}^{\infty}\frac{\mathrm{B}_r}{r!}\mathrm{ad}^r_{\Omega(t)}M$$

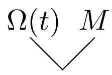
$$= \sum_{r=0}^{\infty}\frac{\mathrm{B}_r}{r!}\mathrm{ad}^r_{\Omega(t)}\sum_{n=0}^{\infty}\frac{1}{n!}$$


$$+\sum_{r=0}^{\infty}\frac{\mathrm{B}_r}{r!}\mathrm{ad}^r_{\Omega(t)}M$$

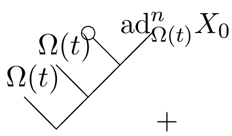
$$= \frac{\mathrm{B}_0}{0!}\Big(\sum_{n=0}^{\infty}\frac{1}{n!}$$


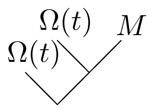
$$+ \quad M\Big)$$

$$+\frac{\mathrm{B}_1}{1!}\Big(\sum_{n=0}^{\infty}\frac{1}{n!}$$


$$+$$


$$\Big)$$

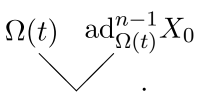
$$+\frac{\mathrm{B}_2}{2!}\Big(\sum_{n=0}^{\infty}\frac{1}{n!}$$


$$+$$


$$\Big)$$

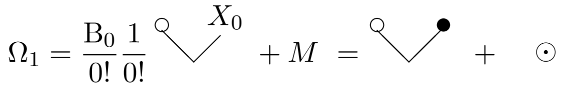
$$+\cdots. \tag{5.6}$$

where

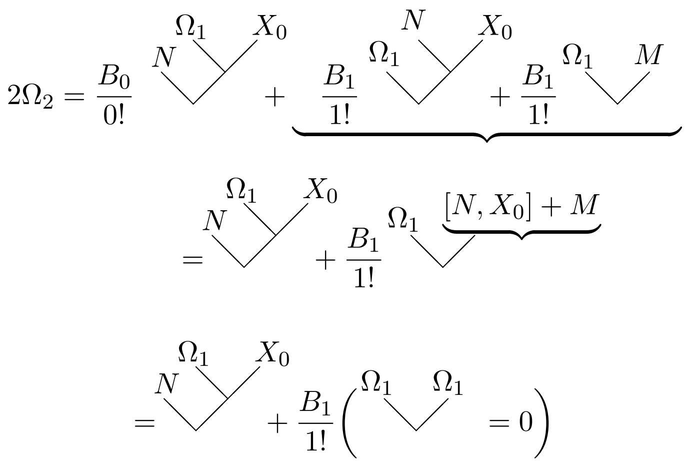
$$\mathrm{ad}^n_{\Omega(t)}X_0 =$$


.

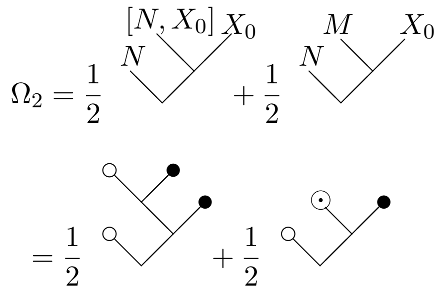
If we compare the coefficients of $t^0, t^1, t^2 \ldots$ we get the following representation



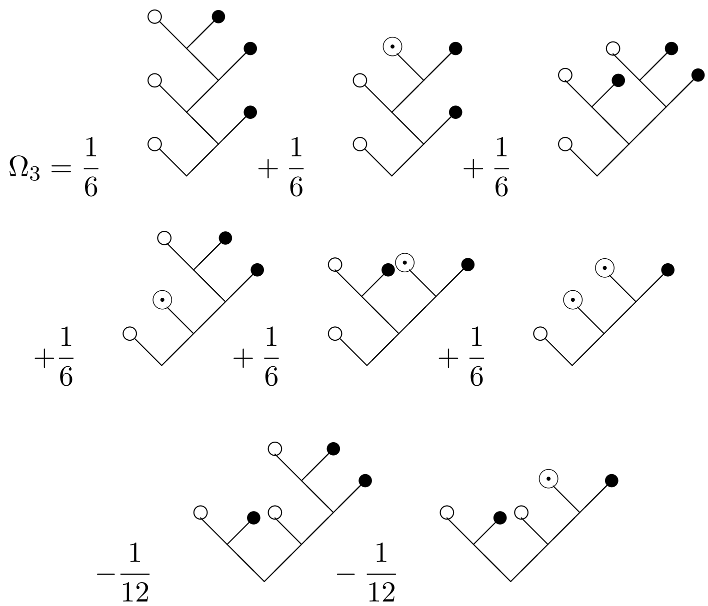
and

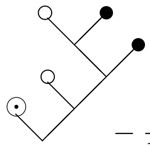
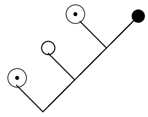


Therefore,

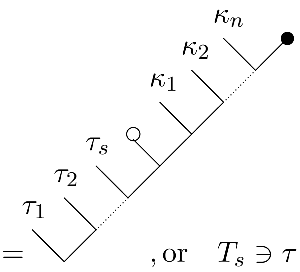


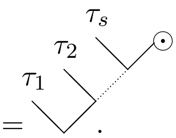
Similarly,



$-\frac{1}{12}$  $-\frac{1}{12}$ 

Clearly, we see that each tree here can be represented in the form

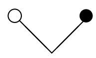
$T_{s,n} \ni \tau =$ 

, or $T_s \ni \tau =$ 

. (5.7)

where $s \in \{0, 1, 2, 3, \ldots\}$, $n \in \{0, 1, 2, 3, \ldots\}$. Here, the trees $\tau_1, \tau_2, \ldots \tau_s$, $\kappa_1, \kappa_2, \ldots \kappa_n$ have been featured earlier in the expansion, where $\tau_i \in T_{p_i}$, and $\kappa_j \in T_{q_j}$, $i = 1, 2, ..., s$, $j = 1, 2..., n$ and $p_1 + p_2 + ... + p_s + q_1 + q_2 + ... + q_n + 1 = r$.

It is also possible to deduce the explicit form of the constant α. Set

$\alpha\Big($  $\Big) = 1$ (5.8)

and

$$\alpha(\odot) = 1. \tag{5.9}$$

Let $\tau \in \mathbb{T}_r, r \in \mathbb{N}$ and suppose that $\alpha(\tau_i)$ and $\alpha(\kappa_j)$ are known for $i = 1, 2, \ldots, s$, $j = 1, 2, \ldots, n$. Then

$$\alpha(\tau) = \frac{1}{r}\frac{\mathrm{B}_s}{s!}\frac{1}{n!}\prod_{i=1}^{s}\alpha(\tau_i)\prod_{j=i}^{n}\alpha(\kappa_j), \quad s, n \in \mathbb{N}, \tag{5.10}$$

where B_s is the sth Bernoulli number. We have the elements set to generate the trees for the Taylor expansion. In the next subsection, we construct the recursive rule (5.7) and (5.10) step by step in an alternative way.

5.2.1 Constructing the elements (5.7) and (5.10)

We may write the dexp equation as

$$\Omega' = \mathrm{dexp}_{\Omega}^{-1}([N, e^{\mathrm{ad}_\Omega} X_0] + M), t \geq 0, \Omega(t_0) = O$$

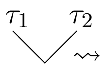
where $e^{\mathrm{ad}_\Omega} = \Sigma_{m=0}^{\infty}(\frac{1}{m!}\mathrm{ad}_{\Omega}^{m})$. We attempt to find the expansion of the solution in Taylor series of the form (5.5). As defined earlier in section 5.2, while constructing trees, we commence by assigning X_0 to a single node, i.e. a *trivial tree*,

$$\bullet \rightsquigarrow X_0.$$

and

$$\circ \rightsquigarrow N, \quad \text{and} \quad \odot \rightsquigarrow M.$$

We define a function $\tau \to H_\tau$ from $\mathbb{T} = \bigcup_{r=1}^{\infty} \mathbb{T}_r$, a subset of binary rooted trees into $n \times n$ matrix functions by letting $H_\bullet = X_0$ and, by induction,



$$\rightsquigarrow [H_{\tau_1}, H_{\tau_2}],$$

where H_{τ_1} and H_{τ_2} are already constructed expansion terms. Let us define

$$\begin{aligned} P_n(t) &= \mathrm{ad}_\Omega^n X_0, \quad n \in \mathbb{Z}_+, \\ Q_s(t) &= \mathrm{ad}_\Omega^s([N, e^{\mathrm{ad}_\Omega} X_0] + M), \quad s \in \mathbb{Z}_+, \end{aligned}$$

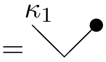
the Taylor expansions of P_n and Q_s can also be written as nested commutators and their linear combinations,

$$\begin{aligned} P_n(t) &= \sum_{k=n}^{\infty} t^k \sum_{\tau \in \mathbb{P}_{n,k}} \beta(\tau) H_\tau, \\ Q_s(t) &= \sum_{l=s}^{\infty} t^l \sum_{\tau \in \mathbb{Q}_{s,l}} \gamma(\tau) H_\tau, \end{aligned}$$

Here, H_τ is an expression constructed from X_0, N and M in the structure of the tree τ, and $\beta(\tau)$ and $\gamma(\tau)$ are scalars. The index sets $\mathbb{P}_{n,k}$ and $\mathbb{Q}_{s,l}$ are formal collections of binary rooted trees. Our goal is to establish the recursive rule from P_n. We commence by establishing the rules to form the sets $\mathbb{P}_{n,k}$, and the coefficient map β. Since $P_0 = X_0$, we obtain $\mathbb{P}_{0,0} = \bullet\, ; \beta(\,\bullet) = 1$

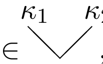
Thus substituting (5.5),

$$\begin{aligned} P_1 &= [\Omega(t), P_0(t)] \\ &= \sum_{m=1}^{\infty} t^m \sum_{\tau \in \mathbb{T}} \alpha(\tau)[H_\tau, X_0], \end{aligned}$$

therefore, for every $k \geq 1$, $\tau \in \mathbb{P}_{1,k} \Leftrightarrow \tau =$  where $\kappa_1 \in \mathbb{T}_k$.

In this case $\beta(\tau) = \alpha(\kappa_1)$. Here, we suppose that we have already determined the sets $\mathbb{P}_{m,k}$ and the underlying coefficient map β, for $m \leq n-1$ and $k \geq m$, we will prove for $m = n$

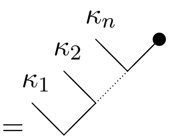
$$\begin{aligned}
P_n &= \mathrm{ad}_\Omega P_{n-1} = [\Omega, P_{n-1}] \\
P_n &= \Big[\sum_{m=1}^{\infty} t^m \sum \alpha(\kappa_1) H_{\kappa_1}, \sum_{k=n-1}^{\infty} t^m \sum \beta(\kappa_2) H_{\kappa_2}\Big] \\
&= \sum_{m=1}^{\infty} \sum_{k=n-1}^{\infty} t^{m+k} \sum \sum \alpha(\kappa_1)\beta(\kappa_2)[H_{\kappa_1}, H_{\kappa_2}],
\end{aligned}$$

therefore, $\tau \in \mathbb{P}_{n,k} \Leftrightarrow \tau \in$ 

, where $\kappa_1 \in \mathbb{T}_{k_1}$, $\kappa_2 \in \mathbb{P}_{n-1,k_2}$, $k_1 + k_2 = k$

$$\beta(\tau) = \alpha(\kappa_1)\beta(\kappa_2)$$

Using induction and noting the fact that $P_0 = X_0$ can now be used to express in terms of trees from $\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_{n-1}$

$\mathbb{P}_{n,k} \ni \tau =$ 

$\Leftrightarrow \kappa_l \in \mathbb{T}_{k,l}, l = 1, 2, \ldots, n, \sum k_l = k, \beta(\tau) = \prod_{i=1}^{n} \alpha(\kappa_i).$

After expressing the P_n in $\mathbb{T}$, we proceed to do the same for the functions Q_s. We commence by noting

$$e^{\mathrm{ad}_\Omega} X_0 = \sum_{n=0}^{\infty} \frac{1}{n!} \mathrm{ad}_\Omega^n X_0 = \sum_{n=0}^{\infty} \frac{1}{n!} P_n$$

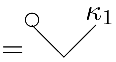
Therefore,

$$\begin{aligned}
Q_0 &= [N, e^{\mathrm{ad}_{\Omega(t)}} X_0] + M = \sum_{r=0}^{\infty} \frac{1}{r}[N, P_n] + M \\
&= \sum_{l=0}^{\infty} t^l \sum_{r=0}^{l} \frac{1}{n!} \sum \beta(\tau)[N, H_\tau] + M.
\end{aligned}$$

hence,

$$\sum \gamma(\tau) H_\tau = \sum_{r=0}^{l} \frac{1}{n!} \sum \beta(\tau)[N, H_\tau] + M.$$

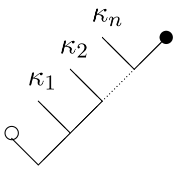
We thus deduce that, for every $l \in \mathbb{Z}_+$,

$\mathbb{Q}_{0,0} \ni \tau =$ 

$+ \quad \odot\,,$

where $\kappa_1 \in \bigcup_{n=0}^{l} \mathbb{P}_{n,l}$ and $\gamma(\tau) = \frac{1}{n!}\beta(\kappa_1)$.

We can reformulate above as,

$\mathbb{Q}_{0,l} \in \tau =$ 

$+ \quad \odot\,,$

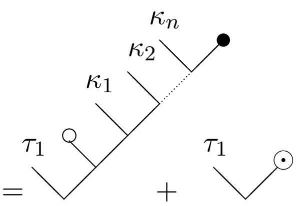
where $\kappa_i \in \mathbb{T}_k, \quad \sum_{i=l}^{r} k_i = l$

$$\beta(\tau) = \frac{1}{n!}\prod_{i=1}^{n} \alpha(\kappa_i).$$

Continuing in the similar manner we get,

$$Q_1(t) = [\Omega(t), Q_0(t)] = \sum_{m=0}^{\infty}\sum_{l=0}^{\infty} t^{m+l} \sum\sum \alpha(\tau_1)\gamma(\tau_2)[H_{\tau_1}, H_{\tau_2}],$$

therefore,

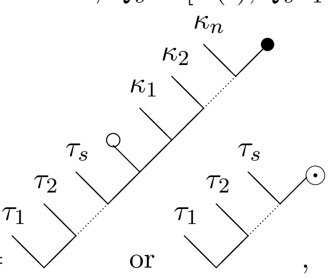
$\mathbb{Q}_{1,l} \ni \tau =$ 

,

where $\quad \kappa_i \in \mathbb{T}_k, \quad q_i + \sum_{i=1}^{s} k_i = l$ and

$$\gamma(\tau) = \frac{1}{n!}\prod_{i=1}^{s} \alpha(\tau_i) \prod_{j=1}^{n} \alpha(\kappa_j). \tag{5.11}$$

proceeding in inductive way using the fact, $Q_s = [\Omega(t), Q_{s-1}(t)], s \in \mathbb{N}$, we deduce that

$\mathbb{Q}_{s,l} \ni \tau =$ 

,

hence we obtained the tree structure (5.7), where $\tau_i \in \mathbb{T}_{q_i}, \kappa_j \in \mathbb{T}_{k_j}, \quad \sum_{i=1}^{s} q_i + \sum_{j=1}^{n} k_j = l$ and

$$\gamma(\tau) = \frac{1}{n!}\prod_{i=1}^{s}\alpha(\tau_i)\prod_{j=1}^{n}\alpha(\kappa_j).$$

Now we work towards the derivation of (5.10). Having derived $Q_s, s \in \mathbb{Z}_+$, we note that

$$\mathrm{dexp}_{\Omega}^{-1}([N, e_{\Omega}^{ad}X_0] + M) = \sum_{s=0}^{\infty}\frac{B_s}{s!}Q_s.$$

Integrating, we deduce from the dexpinv

$$\begin{aligned}\Omega(t) &= \sum_{s=0}^{\infty}\frac{B_s}{s!}\int_{t_0}^{t} Q_s dx = \sum_{s=0}^{\infty}\frac{B_s}{s!}\sum_{m=s}^{\infty}\frac{t^{m+1}}{m+1}\sum\gamma(\tau)H_\tau \\ & \sum_{m=1}^{\infty}\frac{t^m}{m}\sum_{s=0}^{m-1}\frac{B_s}{s!}\sum\gamma(\tau)H_\tau\end{aligned}$$

Thus,

$$\sum\alpha(\tau)H_\tau = \frac{t^m}{m}\sum_{s=0}^{m-1}\frac{B_s}{s!}\sum\gamma(\tau)H_\tau, \quad m \in \mathbb{N}$$

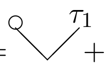
Thus we deduce that $\mathbb{T}_m = \bigcup_{s=0}^{m-1}\mathbb{Q}_{s,m-1}, \quad m \in \mathbb{N}$
Let $\tau \in \mathbb{T}_m, m \in \mathbb{N}$ then $\exists s \in 0, 1, \cdots, m-1$ such that $\tau \in \mathbb{Q}_{s,m-1}$, we deduce

$$\alpha(\tau) = \frac{1}{m}\frac{B_s}{s!}\gamma(\tau).$$

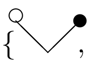
Also, from (5.11), $\gamma(\tau) = \frac{1}{n!}\prod_{i=1}^{s}\alpha(\tau_i)\prod_{j=1}^{n}\alpha(\kappa_j)$. Together which result in the equation (5.10)

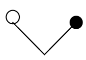
$$\alpha(\tau) = \frac{1}{r}\frac{\mathrm{B}_s}{s!}\frac{1}{n!}\prod_{i=1}^{s}\alpha(\tau_i)\prod_{j=i}^{n}\alpha(\kappa_j), \quad s, n \in \mathbb{N}.$$

Let us take an example, taking $m = 1, \quad \mathbb{T}_1 = \mathbb{Q}_{0,0}$

$\mathbb{Q}_{0,0} =$  $+ \quad \odot, \quad \tau_1 \in \mathbb{P}_{0,0}, \quad \gamma(\tau) = \beta(\tau_1)$

$\mathbb{P}_{0,0} = \{\bullet\}, \beta(\bullet) = 1$ therefore,

$\mathbb{T}_1 = \mathbb{Q}_{0,0} = \{$  $, \quad \odot\}$

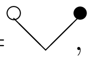
$\alpha\{$  $\} = 1, \quad \alpha\{\odot\} = 1$

This is the set of first order trees. We proceed in the similar way to obtain the next generation trees. Let us jump to the next section to grow the binary rooted trees using this algorithm.

5.2.2 Growing trees using the algorithm

Now we have the general pattern for recursion. Suppose that $\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_{r-1}$ are known and also the coefficient α in these sets is known. To construct $\mathbb{T}_r$ we note that every τ is of the form (5.7) for some $s \in \{0, 1, 2, 3, \ldots, r-1\}$ and $n \in \{0, 1, 2, 3, \ldots, r-1\}$. For every such s and n we consider all the partitions $p_1 + p_2 + ... + p_s + q_1 + q_2 + ... + q_n + 1 = r$. For every partition we construct the tree τ in (5.7) and use (5.10) to determine the coefficient α. The trees which correspond to zero terms would be eliminated. Moreover, some trees can be replaced by linear combinations of other trees. Let us start from $\mathbb{T}_1$

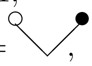
(i) $\mathbb{T}_1$: For $s = 0, n = 0$,

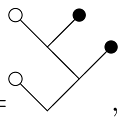
(1) $\tau_1^1 =$ , $\alpha(\tau_1^1) = \frac{1}{1} \cdot 1 \cdot \frac{1}{0!} = 1.$

(2) $\tau_2^1 = \odot$, $\alpha(\tau_2^1) = \frac{1}{1} \cdot 1 \cdot \frac{1}{0!} = 1.$

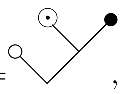
(ii) $\mathbb{T}_2$:

(1) For $s = 0, n = 1$,

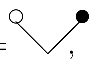
i. When $\kappa_1 =$ ,

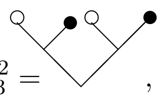
$\tau_1^2 =$ , $\alpha(\tau_1^2) = \dfrac{1}{2} \cdot 1 \cdot \dfrac{1}{1!} = \dfrac{1}{2};$

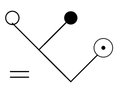
ii. When $\kappa_1 = \odot$,

$\tau_2^2 =$ , $\alpha(\tau_2^2) = \dfrac{1}{2} \cdot 1 \cdot \dfrac{1}{1!} = \dfrac{1}{2};$

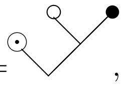
(2) For $s = 1, n = 0$:

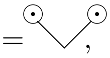
i. When $\tau_1 =$ ,

$\tau_3^2 =$ , vanishing tree, discard.

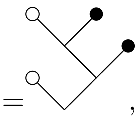
$\tau_4^2 =$ , $\alpha(\tau_4^2) = \dfrac{1}{2}\dfrac{-1}{2} \cdot 1 \cdot \dfrac{1}{0!} = \dfrac{-1}{4}$

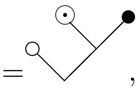
ii. When $\tau_1 = \odot$

$$\tau_5^2 =$$  $$, \quad \alpha(\tau_5^2) = \frac{1}{2} \cdot \frac{-1}{2} \cdot 1 \cdot \frac{1}{0!} = \frac{-1}{4}$$

$$\tau_6^2 =$$  , vanishing tree, discard.

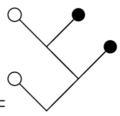
Before we proceed further, let us clean up the set $\mathbb{T}_2$. We clearly see that τ_4^2 is nothing but τ_5^2 with opposite sign. Therefore, both the trees get cancelled as the coefficient is also the same. Therefore, $\mathbb{T}_2$ contains,

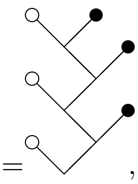
$$\tau_1^2 =$$  $$, \qquad \alpha(\tau_1^2) = \frac{1}{2};$$

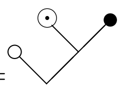
$$\tau_2^2 =$$  $$, \qquad \alpha(\tau_2^2) = \frac{1}{2}.$$

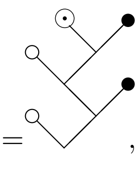
(iii) $\mathbb{T}_3$:

(1) when $s = 0, n = 1$

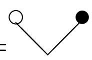
i. When $\kappa_1 =$ 

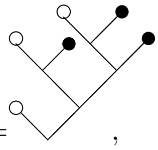
$$\tau_1^3 =$$  $$, \qquad \alpha(\tau_1^3) = \frac{1}{6};$$

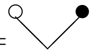
ii. When $\kappa_1 =$ 

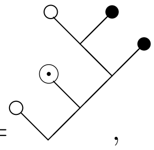
$$\tau_2^3 =$$  $$, \qquad \alpha(\tau_2^3) = \frac{1}{6};$$

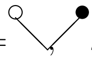
(2) When $s = 0, n = 2$

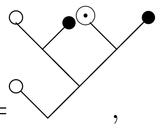
i. When $\kappa_1 = \kappa_2 =$ 

$\tau_3^3 =$  , $\qquad \alpha(\tau_3^3) = \frac{1}{6};$

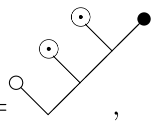
ii. $\kappa_1 = \odot\,, \kappa_2 =$ 

$\tau_4^3 =$  , $\qquad \alpha(\tau_4^3) = \frac{1}{6};$

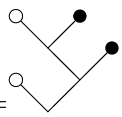
iii. $\kappa_1 =$  , $\kappa_2 = \odot$

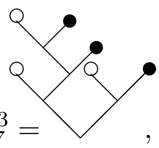
$\tau_5^3 =$  , $\qquad \alpha(\tau_5^3) = \frac{1}{6};$

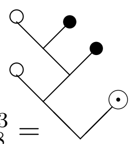
iv. $\kappa_1 = \kappa_2 = \odot$

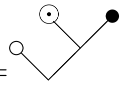
$\tau_6^3 =$  , $\qquad \alpha(\tau_6^3) = \frac{1}{6};$

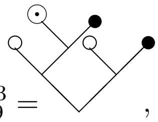
(3) For $s = 1, n = 0$

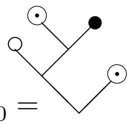
i. $\tau_1 =$ 

$\tau_7^3 =$  , $\qquad \alpha(\tau_7^3) = -\frac{1}{12};$

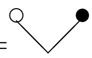
$\tau_8^3 =$  , $\qquad \alpha(\tau_8^3) = -\frac{1}{12};$

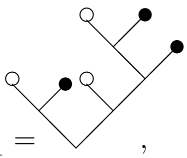
ii. $\tau_1 =$ 

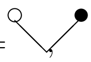
$\tau_9^3 =$  , $\alpha(\tau_9^3) = -\frac{1}{12}$;

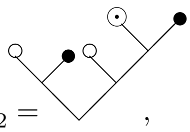
$\tau_{10}^3 =$  , $\alpha(\tau_{10}^3) = -\frac{1}{12}$;

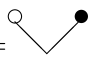
(4) For $s = 1, n = 1$

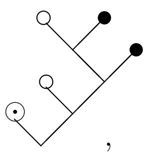
i. $\tau_1 = \kappa_1 =$ 

$\tau_{11}^3 =$  , $\alpha(\tau_{11}^3) = -\frac{1}{6}$;

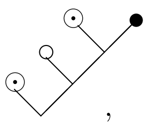
ii. $\tau_1 =$  , $\kappa_1 = \odot$

$\tau_{12}^3 =$  , $\alpha(\tau_{12}^3) = -\frac{1}{6}$;

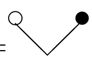
iii. $\tau_1 = \odot$, $\kappa_1 =$ 

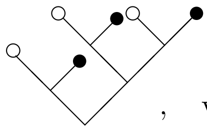
$\tau_{13}^3 =$  , $\alpha(\tau_{13}^3) = -\frac{1}{6}$;

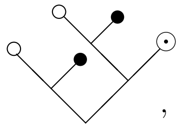
iv. $\tau_1 = \kappa_1 = \odot$

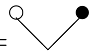
$\tau_{14}^3 =$  , $\alpha(\tau_{14}^3) = -\frac{1}{6}$;

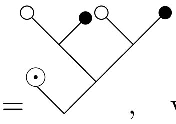
(5) For $s = 2, n = 0$

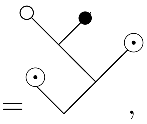
i. When $\tau_1 = \tau_2 =$ 

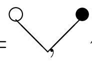
$\tau^3_{15} =$  , vanishing tree, discard;

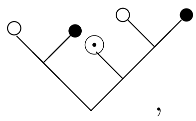
$\tau^3_{16} =$  , $\alpha(\tau^3_{16}) = \dfrac{1}{36}$;

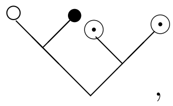
ii. $\tau_1 = \odot$, $\tau_2 =$ 

$\tau^3_{17} =$  , vanishing tree, discard;

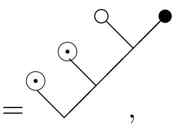
$\tau^3_{18} =$  , $\alpha(\tau^3_{18}) = \dfrac{1}{36}$;

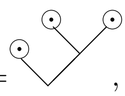
iii. $\tau_1 =$  , $\tau_2 = \odot$

$\tau^3_{19} =$  , $\alpha(\tau^3_{19}) = \dfrac{1}{36}$;

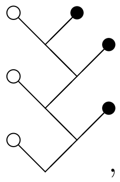
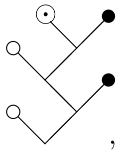
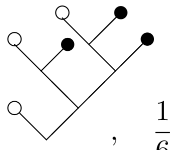
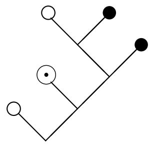
$\tau^3_{20} =$  , vanishing tree, discard;

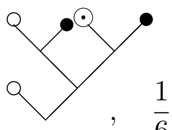
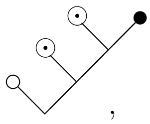
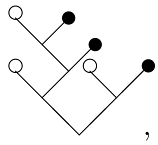
iv. $\tau_1 = \tau_2 = \odot$

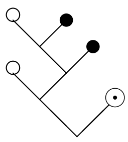
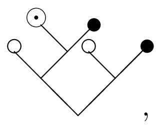
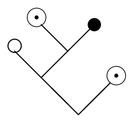
$\tau^3_{21} =$  , $\alpha(\tau^3_{21}) = \dfrac{1}{36}$;

$\tau^3_{22} =$  , vanishing tree, discard.

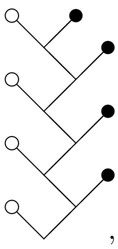
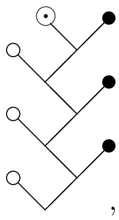
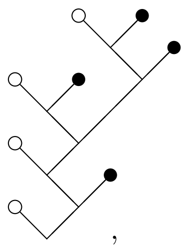
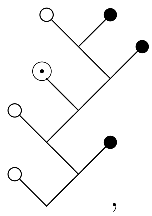
Cleaning up the set $\mathbb{T}_3$. We clearly see that τ^3_{11} is nothing but τ^3_7 with opposite sign. The two trees can be aggregated into τ^3_7 say, with the coefficient replaced by $\alpha(\tau^3_7) - \alpha(\tau^3_{11})$. Similarly, $\tau^3_{12} = -\tau^3_9$, $\tau^3_{13} = -\tau^3_8$ and $\tau^3_{14} = -\tau^3_{10}$. Proceeding further we see that τ^3_{15}, τ^3_{17}, τ^3_{20}, τ^3_{22} are vanishing whereas, τ^3_{19} is nothing but τ^3_{16} with opposite sign also τ^3_{21} is negative of τ^3_{18}. Therefore, simplifying the above generation, the trees having same coefficient get cancelled and the others get convoluted with adding or subtracting the respective coefficients. After trivial rotations (corresponding to commutation) we obtain the set $\mathbb{T}_3$,

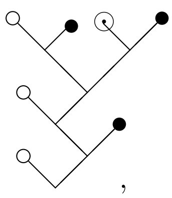
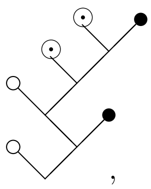
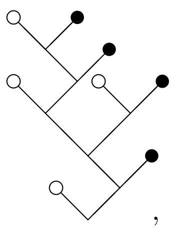
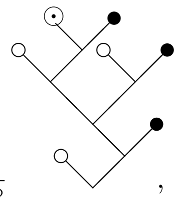
$\frac{1}{6}$  , $\frac{1}{6}$  , $\frac{1}{6}$  , $\frac{1}{6}$ 

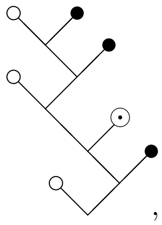
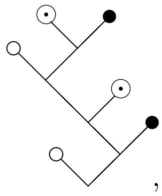
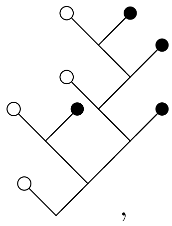
$\frac{1}{6}$  , $\frac{1}{6}$  , $\frac{1}{12}$  ,

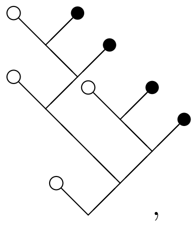
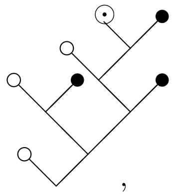
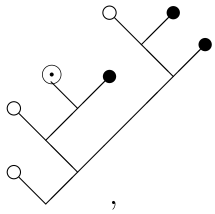
$\frac{1}{12}$  , $\frac{1}{12}$  , $\frac{1}{12}$  .

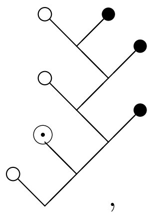
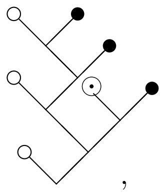
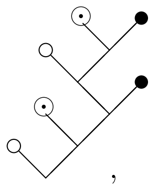
Next we generate the set $\mathbb{T}_4$ of fourth order trees. We get 102 trees with all the possible combinations but these get reduced with the simplifications. Some of trees are zero trees and few get cancelled or added up to the others because of anti symmetry and Jacobi identity. Also, when we have $s = 3$, we get zero coefficient with respect to those trees because the Bernoulli's number B_3 equals zero. Hence we lose the trees with coefficient containing B_3. After all the possible simplifications we are left with the following thirty eight trees with respective coefficients. Therefore, $\mathbb{T}_4$ contains,

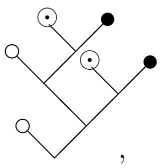
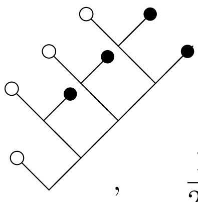
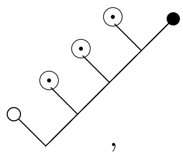
$\frac{1}{48}$  , $\frac{1}{48}$  , $\frac{1}{48}$  , $\frac{1}{48}$  ,

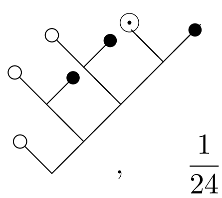
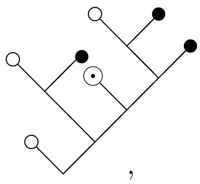
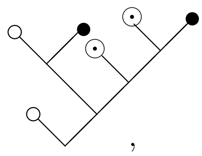
$\frac{1}{48}$  , $\frac{1}{48}$  , $\frac{1}{48}$  , $\frac{1}{48}$  ,

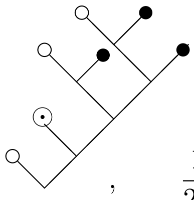
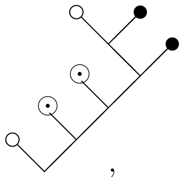
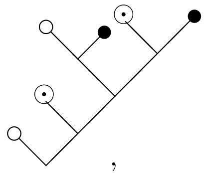
$\frac{1}{48}$  , $\frac{1}{48}$  , $\frac{1}{16}$  ,

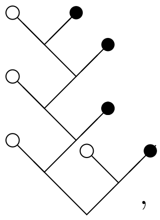
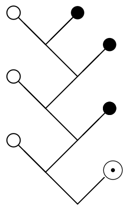
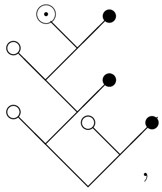
$\frac{1}{16}$  , $\frac{1}{16}$  , $\frac{1}{16}$  ,

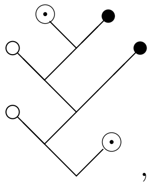
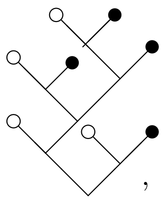
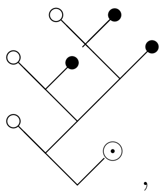
$\frac{1}{16}$  , $\frac{1}{16}$  , $\frac{1}{16}$  ,

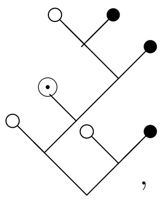
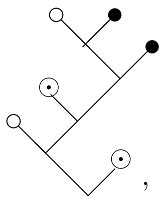
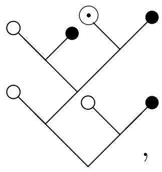
$\frac{1}{16}$  , $\frac{1}{24}$  , $\frac{1}{24}$  ,

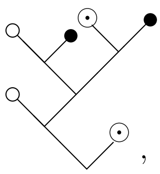
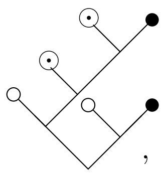
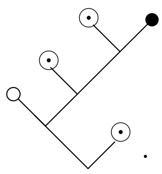
$\frac{1}{24}$  , $\frac{1}{24}$  , $\frac{1}{24}$  ,

$\frac{1}{24}$  , $\frac{1}{24}$  , $\frac{1}{24}$  ,

$\frac{1}{24}$  , $\frac{1}{24}$  , $\frac{1}{24}$  ,

$\frac{1}{24}$  , $\frac{1}{24}$  , $\frac{1}{24}$  ,

$\frac{1}{24}$  , $\frac{1}{24}$  , $\frac{1}{24}$  ,

$\frac{1}{24}$  , $\frac{1}{24}$  , $\frac{1}{24}$  .

Translating the trees in terms of commutators, the Taylor expansion of $\Omega(t)$ becomes,

$$
\begin{aligned}
\Omega(t) \;=\;& t([N,X]+M)\\
&+t^2(\frac{1}{2}[N,[[N,X],X]]+\frac{1}{2}[N,[M,X]]\\
&+t^3(\frac{1}{6}[N,[[N,[[N,X],X]],X]]+\frac{1}{6}[N,[[N,[M,X]],X]]\\
&+\frac{1}{6}[N,[[N,X],[[N,X],X]]]+\frac{1}{6}[N,[M,[[N,X],X]]]\\
&+\frac{1}{6}[N,[[N,X],[M,X]]]+\frac{1}{6}[N,[M,[M,X]]]\\
&+\frac{1}{12}[[N,[[N,X],X]],[N,X]]+\frac{1}{12}[[N,[M,X]],[N,X]]\\
&+\frac{1}{12}[[N,[[N,X],X]],M]+\frac{1}{12}[[N,[M,X]],M])\\
&+t^4(\frac{1}{24}[N,[[N,[[N,[[N,X],X]],X]],X]]+\frac{1}{24}[N,[[N,[[N,[M,X]],X]],X]]\\
&+\frac{1}{24}[N,[[N,[[N,X],[[N,X],X]]],X]]+\frac{1}{24}[N,[[N,[M,[[N,X],X]]],X]]\\
&+\frac{1}{24}[N,[[N,[[N,X],[M,X]]],X]]+\frac{1}{24}[N,[[N,[M,[M,X]]],X]]\\
&+\frac{1}{48}[N,[[[N,[[N,X],X]],[N,X]],X]]+\frac{1}{48}[N,[[[N,[M,X]],[N,X]],X]]\\
&+\frac{1}{48}[N,[[[N,[[N,X],X]],M],X]]+\frac{1}{48}[N,[[[N,[M,X]],M],X]]\\
&+\frac{1}{16}[N,[[N,X],[[N,[[N,X],X]],X]]]+\frac{1}{16}[N,[[N,[[N,X],X]],[[N,X],X]]]\\
&+\frac{1}{16}[N,[[N,X],[[N,[M,X]],X]]]+\frac{1}{16}[N,[[N,[M,X]],[[N,X],X]]]\\
&+\frac{1}{16}[N,[M,[[N,[[N,X],X]],X]]]+\frac{1}{16}[N,[[N,[[N,X],X]],[M,X]]]\\
&+\frac{1}{16}[N,[M,[[N,[M,X]],X]]]+\frac{1}{16}[N,[[N,[M,X]],[M,X]]]\\
&+\frac{1}{24}[N,[[N,X],[[N,X],[[N,X],X]]]]+\frac{1}{24}[N,[M,[M,[M,X]]]]\\
&+\frac{1}{24}[N,[[N,X],[[N,X],[M,X]]]]+\frac{1}{24}[N,[[N,X],[M,[[N,X],X]]]]\\
&+\frac{1}{24}[N,[[N,X],[M,[M,X]]]]+\frac{1}{24}[N,[M,[[N,X],[[N,X],X]]]]\\
&+\frac{1}{24}[N,[M,[M,[[N,X],X]]]]+\frac{1}{24}[N,[M,[[N,X],[M,X]]]]\\
&+\frac{1}{24}[[N,[[N,[[N,X],X]],X]],[N,X]]+\frac{1}{24}[[N,[[N,[[N,X],X]],X]],M]\\
&+\frac{1}{24}[[N,[[N,[M,X]],X]],[N,X]]+\frac{1}{24}[[N,[[N,[M,X]],X]],M]\\
&+\frac{1}{24}[[N,[[N,X],[[N,X],X]]],[N,X]]+\frac{1}{24}[[N,[[N,X],[[N,X],X]]],M]\\
&+\frac{1}{24}[[N,[M,[[N,X],X]]],[N,X]]+\frac{1}{24}[[N,[M,[[N,X],X]]],M]
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{1}{24}[[N,[[N,X],[M,X]]],[N,X]]+\frac{1}{24}[[N,[[N,X],[M,X]]],M]\\
&+\frac{1}{24}[[N,[M,[M,X]]],[N,X]]+\frac{1}{24}[[N,[M,[M,X]]],M])\\
&+\cdots .
\end{aligned}
$$

Once we obtain a tree formulation, translating it into commutators costs less labour as compared to the complexity of manual computation of the terms. Subsequent to converting the trees into mathematical expressions, we plot the error graph. Truncating the expansion up to fourth order terms, the error graph of the solution as compared to the MATLAB ode45 solver with built-in parameters, is shown in Figure 5.3. The error plot is generated by comparing against the theoretically expected error of $O((\Delta t)^4)$. The experiments were performed on random initial matrices.

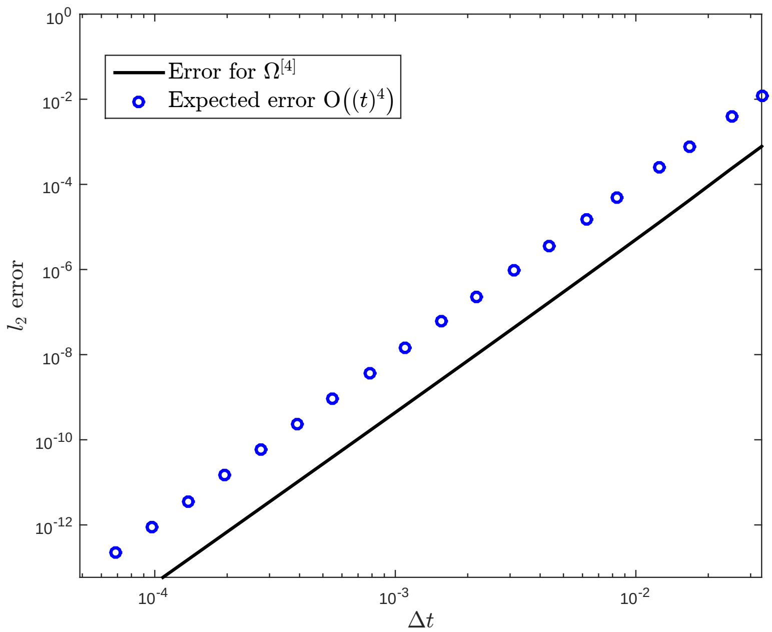


Figure 5.3: Global error on logarithmic scale across an interval [0,1] with different time steps, after truncating the Taylor expansion up to fourth order terms.

We thus observe that Lie group methods have the generality to be applied to various isospectral flows. Although the matrix system (4.1) appears more complex and leads to the tri-colour leaves; it has been possible to formulate the explicit recursive rule. This matrix system has two types of trees and an interplay between them, which makes it more difficult, yet we successfully develop a step by step algorithm and define a recurrence formula that enables us to compute the trees and coefficients explicitly. Figure 5.2 shows that the discretisation of the gdb equations using Magnus expansion preserves the eigenvalues of the solution matrix. Also, it is observed in Figure 5.3 that the Lie group method using Magnus expansion is a fourth order method; here by fourth order we mean the truncation of $\Omega(t)$ upto the fourth power of t. By employing the shorthand of binary rooted trees for expansion terms, the computation is made affordable. This also lays a foundation to the explicit representation of the solution of the generalised double bracket flow.

Chapter 6

Conclusion

In this thesis we considered solving isospectral flows, the ordinary differential equations of the form

$$X^{'} = [B(X), X], \quad t \geq 0, \quad X(0) = X_0,$$

possessing the characteristic to leave the spectrum of the initial matrix X_0 invariant in the solution space X. Evidently, these flows emerge in a number of remarkable applications ranging from classical mechanics to linear algebra. Traditional numerical methods do not preserve the isospectrality of the system. Therefore, we emphasise on solving the isospectral flows using Lie group methods which preserves the isospectrality of the solution space and gives the desired structure of the solution with large time steps. The emphasis is also given on presenting the terms using shorthand of binary rooted trees that enables us to derive a recursive relation to generate text trees and calculate their coefficients. That also helps us to reduce the manual computational cost. Also, using anti symmetry and Jacobi identity to reduce the trees to the independent terms, further reduces the cost.

We observe in chapter 3 that BI equations consist of phenomenal properties and hence, motivate us to expand the equations. It has been seen that these equations have Lie-Poisson structure. The comparison of the error of the solution against ode45 after truncating up to third order and fourth order terms gives the desired result. Discretisation of the BI equations using Magnus expansion preserves the eigenvalues of the solution matrix. This favourable behavior of the Magnus method is to be expected from the principles underlying our approach. The shorthand of binary rooted trees for expansion terms, makes the computation affordable. This also lays a foundation to the explicit representation of the solution of the BI system. Motivated by the promising results of Lie group methods for BI equations, we applied the procedure on generalised double bracket flow and observed the favourable results. Though the flow in chapter 5 is much more complicated and has to result in tri-colour leaves while constructing trees, yet one observes that it has been possible to derive an explicit representation of the solution using Magnus expansion and binary rooted trees and preserve the desired structure leaving the eigenvalues invariant.

Whereas in chapter 4, we observe the dynamical structure of the flow. It is observed that the system has Hopf bifurcation that gives birth to the limit cycles.

There are a number of issues that can be addressed in future. An attempt can be made to formulate the rule to count the trees. It has been known that the number of trees having $2n$ nodes is calculated using *Witt–Birkhoff theorem* $\frac{1}{2n}\sum_{d|n}\mu\left(\frac{n}{d}\right)\binom{2d}{d}$, where μ is *Möbius function*, and the number of ordered trees of degree $n+1$ is given by the *Catalan numbers* $C(n) = \frac{(2n)!}{n!(n+1)!}$ (see [MKK03] [Ise02] [Bou08]). Upper limit of independent trees for BI equations can be known using $\frac{1}{2n}\sum_{d|n}\mu\left(\frac{n}{d}\right)\binom{2d}{d}$. Whereas in the case of gdbf, we have two types of trees, and an interplay between them, so it does not seem to be easy to determine the number of independent trees in each iteration but might be possible to formulate the rule. Lie group methods are geometric integrators and arise in a number of applications. Numerical implementation of Lie group methods has been seen in [Ise02] [Zan98] [IMKNZ99] [Ise99]. In this thesis the technique has been extended to nonlinear cases and we believe that this would have generality to be applied to a variety of problems in the same class. Also, the generalised form of BI equations $X' = [N, X^m], \quad t \geq 0, \quad X(0) = X_0 \in \mathrm{Sym}(n), \ N \in \mathfrak{so}(n)$, for different values of m can be studied, which for sure would have complexity in handling large number of brackets in the expansion depending upon how large is the power of X.

Bibliography

[BBI$^+$09] A. M. Bloch, V. Brnzanescu, A. Iserles, J. E. Marsden, and T. S. Ratiu, *A Class of Integrable Flows on the Space of Symmetric Matrices*, Communications in Mathematical Physics **290** (2009), 399–435. (Cited on page 38).

[BI05] A. M. Bloch and A. Iserles, *Aspects of generalized double-bracket flows*, Proc. Centre de Recherche Montreal, American Mathematical Society (2005), 65–76. (Cited on page 13).

[BI06] A.M. Bloch and A. Iserles, *On an isospectral Lie–Poisson system and its Lie-algebra*, Found. Comput. Math. (2006), 121–144. (Cited on pages 33 and 38).

[Blo90] A. M. Bloch, *Steepest descent, linear programming and Hamiltonian flows*, Contemp. Math. AMS **114** (1990), 77–88. (Cited on page 52).

[Bou08] N. Bourbaki, *Lie groups and Lie algebras*, vol. 7, Springer Science & Business Media, 2008. (Cited on page 92).

[Bro91] R.W. Brockett, *Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems*, Linear Algebra and its applications **146** (1991), 79–91. (Cited on pages 13 and 52).

[Cas04] F. Casas, *Numerical integration methods for the double-bracket flow*, Journal of Computational and Applied Mathematics **166** (2004), no. 2, 477 – 495. (Cited on pages 13 and 33).

[CD89] M. Chu and K. R. Driessel, *Can real symmetric Toeplitz matrices have arbitrary real spectra?*, Tech. report, Idaho State University, 1989. (Cited on pages 13 and 20).

[CD90] M.T. Chu and K.R. Driessel, *The projected gradient method for least squares matrix approximations with spectral constraints*, SIAM Journal on Numerical Analysis **27** (1990), no. 4, 1050–1060. (Cited on pages 13 and 52).

[Chu84] M. Chu, *The generalized Toda flow, the QR algorithm and the center manifold theory*, SIAM Journal on Algebraic Discrete Methods **5** (1984), no. 2, 187–201. (Cited on pages 13 and 20).

[CIZ97] M.P. Calvo, A. Iserles, and A. Zanna, *Numerical solution of isospectral flows*, Mathematics of computation **66** (1997), no. 220, 1461–1486. (Cited on pages 14, 20, 21, and 34).

[CIZ99] M P. Calvo, A. Iserles, and A. Zanna, *Conservative methods for the Toda lattice equations*, IMA journal of numerical analysis **19** (1999), no. 4, 509–523. (Cited on pages 13, 14, 20, and 34).

[Dem90] J. W. Demmel, *Matrix Computations; Second Edition (Gene Golub and Charles F. Van Loan)*, SIAM Review **32** (1990), no. 4, 690–691. (Cited on page 21).

[DNT83] P. Deift, T. Nanda, and C. Tomei, *Ordinary differential equations and the symmetric eigenvalue problem*, SIAM Journal on Numerical Analysis **20** (1983), no. 1, 1–22. (Cited on pages 13 and 20).

[DRTW91] P. Deift, S. Rivera, C. Tomei, and D. Watkins, *A monotonicity property for Toda-type flows*, SIAM Journal on Matrix Analysis and Applications **12** (1991), no. 3, 463–468. (Cited on pages 13 and 20).

[DRVV94] Luca Dieci, Robert D. Russell, and Erik S. Van V., *Unitary integrators and applications to continuous orthonormalization techniques*, SIAM Journal on Numerical Analysis **31** (1994), no. 1, 261–281. (Cited on page 21).

[DVV99] Luca Dieci and Erik S Van V, *Computation of orthonormal factors for fundamental solution matrices*, Numerische Mathematik **83** (1999), no. 4, 599–620. (Cited on page 21).

[Fla74] H. Flaschka, *The Toda lattice I*, Phys. Rev. B9 (1974). (Cited on page 19).

[FNO87] S. Friedland, J. Nocedal, and M. Overton, *The formulation and analysis of numerical methods for inverse eigenvalue problems*, SIAM Journal on Numerical Analysis **24** (1987), no. 3, 634–667. (Cited on pages 13 and 20).

[GVL89] G.H. Golub and C.F. Van Loan, *Matrix Computations 2nd Edn Johns Hopkins University Press*, 1989. (Cited on page 21).

[Har69] F. Harary, *Graph Theory*, Addison-Wesley, Reading MA, 1969. (Cited on page 28).

[Hau06] F. Hausdorff, *Die symbolische Exponentialformel in der Gruppentheorie*, Berichte der Sächsischen Akademie der Wißenschaften, Math. Phys. Klasse **58** (1906), 19–48. (Cited on page 28).

[HJ91] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991. (Cited on page 34).

[IMKNZ99] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett, and A. Zanna, *Lie-group methods*, Acta Numerica (1999). (Cited on pages 17, 21, 52, 68, and 92).

[IN99] A. Iserles and S.P. Nørsett, *On the Solution of linear differential equations in Lie groups*, Phil. Trans Royal Soc. A **357** (1999), 983–1019. (Cited on pages 27, 28, 37, 40, and 71).

[Ise99] A. Iserles, *Expansions that grow on trees*, Notices Amer. Math. Soc **49** (1999), 430–440. (Cited on pages 17, 27, 29, and 92).

[Ise02] A. Iserles, *On the discretization of double-bracket flows*, Foundations of Computational Mathematics **2** (2002), no. 3, 305–329. (Cited on pages 13, 33, 36, 37, 49, 50, 52, 53, and 92).

[Kau16] A. Kaur, *On solving an isospectral flow*, Journal of Computational and Applied Mathematics **308** (2016), 263 – 275. (Cited on page 33).

[Lag91] J. Lagarias, *Monotonicity properties of the Toda flow, the QR-flow, and subspace iteration*, SIAM Journal on Matrix Analysis and Applications **12** (1991), no. 3, 449–462. (Cited on pages 13 and 20).

[Lax68] P. D. Lax, *Integrals of Nonlinear Equations of Evolution and Solitary Waves*, Commun. Pure Appl. Math. **21** (1968), 467–490. (Cited on page 17).

[LD94] E. S. V. Vleck L. Dieci, R. D. Russell, *Unitary Integrators and Applications to Continuous Orthonormalization Techniques*, SIAM Journal on Numerical Analysis **31** (1994), no. 1, 261–281. (Cited on pages 21 and 68).

[LT06] L. Li and C. Tomei, *C.: The complete integrability of a Lie-Poisson system proposed by Bloch and Iserles*, Intern. Math. Res. Notes 64949 (2006). (Cited on page 38).

[Mag54] W. Magnus, *On the exponential solution of differential equations for a linear operator*, Comm. Pure Appl. Math. **7** (1954), 649–673. (Cited on pages 27, 28, and 35).

[Man76] S.V. Manakov, *Note on the integration of Euler's equations of the dynamics of an n-dimensional rigid body*, Funct. Anal. Appl. **10** (1976), 328–329. (Cited on page 34).

[MKK03] H. Munthe-Kaas and S. Krogstad, *On enumeration problems in liebutcher theory*, Future Generation Computer Systems **19** (2003), no. 7, 1197 – 1205, Selected papers on Theoretical and Computational Aspects of Structural Dynamical Systems in Linear Algebra and Control. (Cited on page 92).

[MM12] J. E. Marsden and M. McCracken, *The Hopf bifurcation and its applications*, vol. 19, Springer Science & Business Media, 2012. (Cited on page 65).

[Nan82] T. Nanda, *Isospectral flows on band matrices*, Ph.D. thesis, New York University, 1982. (Cited on pages 13 and 20).

[Sac64] R. J. Sacker, *On invariant surfaces and bifurcation of periodic solutions of ordinary differential equations*, Tech. report, DTIC Document, 1964. (Cited on page 65).

[Sym82] W. W. Symes, *The QR algorithm and scattering for the finite nonperiodic Toda lattice*, Physica D Nonlinear Phenomena **4** (1982), 275–280. (Cited on pages 13, 19, and 20).

[Tod81] M. Toda, *Theory of Nonlinear Lattices*, Springer-Verlag, Berlin, 1981. (Cited on pages 13, 18, and 19).

[Tod89] ______, *Nonlinear Waves and Solitons*, Mathematics and Its Applications (Japanese Series) (1989). (Cited on pages 13 and 18).

[VDM85] J. Van Der Meer, *The hamiltonian Hopf bifurcation*, Springer, 1985. (Cited on page 65).

[Zan97] A. Zanna, *Lie-group methods for isospectral flows*, University of Cambridge Department of Applied Mathematics and Theoretical Physics-Report-DAMTP NA (1997). (Cited on page 52).

[Zan98] A. Zanna, *On the numerical solution of isospectral flows*, Ph.D. thesis, University of Cambridge, 1998. (Cited on pages 14, 17, 20, 34, and 92).